



# Informatica® Cloud Application Integration

## October 2023

# 監視

Informatica Cloud Application Integration 監視  
October 2023

© 著作権 Informatica LLC 1993, 2023

発行日: 2023-11-28

# 目次

<b>序文</b> .....	6
<b>第 1 章 : アプリケーション統合コンソール</b> .....	7
<b>第 2 章 : プロセスサーバーの設定</b> .....	8
テナントの詳細.....	8
プロパティ.....	9
サマリ.....	9
サーバーのプロパティ.....	10
ログの.....	17
監視.....	19
受信キュー.....	23
ディスパッチサービス.....	24
ディスパッチサービスのコンポーネント.....	25
カスタムディスパッチ設定の作成.....	26
ディスパッチサービスの設定.....	27
ディスパッチ設定の監視.....	28
ディスパッチ設定の例.....	29
ディスパッチサービスのトラブルシューティング.....	33
ディスパッチサービスに関連するその他のコントロール.....	34
ストレージ.....	34
スケジュールされたメンテナンス.....	35
システムサービス.....	36
警告サービス.....	36
電子メールサービス.....	37
シェルサービス.....	39
メッセージサービス.....	39
データソースサービス.....	43
<b>第 3 章 : デプロイされたアセット</b> .....	47
ビジネスプロセスアーカイブのデプロイ.....	47
コントリビューションについて.....	47
コントリビューションの詳細.....	50
デプロイされたプロセス.....	52
リソース.....	52
カタログリソースの表示.....	53
インデックス付きプロパティ.....	55
URN マッピング.....	56

<b>第4章：プロセススケジュール</b>	<b>59</b>
新しいプロセススケジュールの作成	60
実行スケジュールの管理および変更	62
<b>第5章：プロセス</b>	<b>63</b>
プロセスフィルタ	64
簡易プロセスフィルタ	64
詳細プロセスフィルタ	67
[プロセスビューの詳細] ページ	76
開始ステップのプロパティ	77
フォールトの詳細	77
[アクティブなプロセスの詳細] ページ	79
[プロセスバージョンのリスト] ページ	88
[デプロイされたプロセスバージョンの詳細] ページ	89
プロセスバージョンのライフサイクルについて	91
プロセスバージョンの更新	92
プロセスバージョンの設定（オフラインまたはオンライン）	92
ログレベルの設定	92
プロセスバージョンの永続タイプの設定	93
リカバリタイプの呼び出し	94
プロセスインスタンスの保持期間	94
デプロイされたプロセスの詳細グラフ	95
プロセスの再開	95
アプリケーション統合コンソールからのプロセスの再開	96
アプリケーション統合からのプロセスの再開	96
<b>第6章：プロセスサーバーの正常性</b>	<b>98</b>
パフォーマンス	98
アラート	99
統計	99
<b>第7章：プロセスメトリック</b>	<b>101</b>
アラームキューメトリック	101
受信キューメトリック	102
<b>第8章：API</b>	<b>104</b>
API の表示	104
API の非アクティブ化	105
API の有効化	107
<b>第9章：接続</b>	<b>109</b>
リスナベース接続でのイベントソースの開始と停止	110

<b>第 10 章 : ガイド</b> .....	112
[ガイド結果] ページ.....	113
<b>第 11 章 : ログ</b> .....	115
デプロイメントログ.....	115
サーバーログ.....	116
プロセスサーバーのログレベルの設定.....	118
プロセスサーバーログの削除.....	119

# 序文

『監視』で、アプリケーション統合コンソールでアセットを監視および管理する方法を確認します。

# 第 1 章

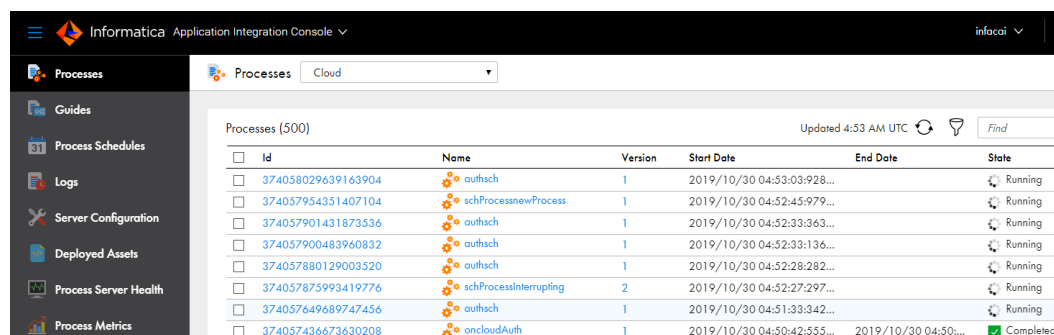
## アプリケーション統合コンソール

アプリケーション統合コンソールは、Informatica Cloud Server と、新しい実行環境で作成した Secure Agent のどちらで実行する場合も、プロセスを監視および管理するためのオプションを提供します。

Informatica Cloud でアプリケーション統合コンソールを使用する場合、ホームページは次に示すような画面です。ただし、使用できるオプションは次に応じて変わります。

- ユーザーアカウントの権限。
- エージェントとクラウドサーバーのどちらを選択するか。
- 使用する製品バージョンに基づいたその他のデプロイメントオプション。

**注:** 使用できるオプションの種類は状況に応じて大きく変わるため、このオンラインヘルプに含まれるトピックが当てはまらない場合もあります。



The screenshot shows the Informatica Application Integration Console interface. On the left is a navigation sidebar with options: Processes, Guides, Process Schedules, Logs, Server Configuration, Deployed Assets, Process Server Health, and Process Metrics. The main area displays a table of processes under the 'Processes' tab, which is set to 'Cloud'. The table has columns for Id, Name, Version, Start Date, End Date, and State. It lists several processes, mostly in a 'Running' state, and one in a 'Completed' state.

Id	Name	Version	Start Date	End Date	State
374058029639163904	authsch	1	2019/10/30 04:53:03:928...		Running
374057954351407104	schProcessnewProcess	1	2019/10/30 04:52:45:979...		Running
374057901431873536	authsch	1	2019/10/30 04:52:33:363...		Running
374057900483960832	authsch	1	2019/10/30 04:52:33:136...		Running
374057880129003520	authsch	1	2019/10/30 04:52:28:282...		Running
374057875993419776	schProcessInterrupting	2	2019/10/30 04:52:27:297...		Running
374057649689747456	authsch	1	2019/10/30 04:51:33:342...		Running
374057436673630208	oncloudAuth	1	2019/10/30 04:50:42:555...	2019/10/30 04:50:...	Completed

## 第 2 章

# プロセスサーバーの設定

[サーバーの環境設定] ページには次の詳細が表示されます。

- テナントまたはプロセスサーバーのプロパティ
- 監視設定
- キュー設定
- ストレージおよびスケジュールメンテナンス設定
- システムサービス設定

## テナントの詳細

[テナントの詳細] セクションには、テナントに関する情報が表示されます。このページは、クラウドサーバーでのみ使用できます。

### ▼ Tenant Detail

Tenant Context Id: 5qBxXt1yz3Gj9VI1HsdT41  
Tenant Name: infai  
Activity Execution Limit: 10000  
Admin Groups:  
Service Consumer Groups:  
Console URL: activevos/5qBxXt1yz3Gj9VI1HsdT41  
SOAP services URL: active-bpel/services/5qBxXt1yz3Gj9VI1HsdT41  
REST services URL: active-bpel/services/REST/5qBxXt1yz3Gj9VI1HsdT41

項目	説明
テナントコンテキスト ID	組織 ID。
テナント名	説明的な名前。

項目	説明
アクティビティ実行制限	<p>実行できるビジネスプロセス実行言語（BPEL）アクティビティの最大数。このオプションを使用して、無限ループと多数のリソースを使用しているプロセスを一時停止または終了します。デフォルトは 10,000 です。</p> <p>プロセスをパブリッシュすると、アプリケーション統合はプロセスの実行可能表現である BPEL を生成します。プロセスの詳細ビューで、プロセスの BPEL 表現を確認できます。</p> <p>アクティビティ実行制限値は、プロセスステップレベルではなく、BPEL アクティビティレベルでのカウントを定義します。1 つのプロセスステップが複数の BPEL アクティビティに変換される可能性があります。したがって、Process Designer でのステップ数が 10,000 未満のプロセスは、10,000 を超える BPEL アクティビティに変換され、デフォルトのアクティビティ実行制限に達する可能性があります。BPEL アクティビティの総数を確認するには、プロセス実行ログに Executing [というキーワードが登場する回数を数えることができます。</p> <p>プロセスが最大レベルに達すると、アプリケーション統合は、プロセスに設定された永続性レベルに基づいてプロセスを一時停止または終了します。永続性レベルが【フル】または【永続】に設定されている場合、アプリケーション統合はプロセスを一時停止します。永続性レベルが【なし】、【簡易】、または【最終】に設定されている場合、アプリケーション統合はプロセスを終了します。</p> <p>一時停止したプロセスは手動で再開する必要があります。一時停止されたプロセスを再開すると、アクティビティの実行数はゼロにリセットされます。この制限は、Secure Agent を使用してプロセスを実行すると回避できます。</p>
管理グループ	テナントに定義された管理グループの名前。
サービスコンシューマグループ	テナントに定義されたサービスコンシューマグループの名前。
コンソールの URL	テナントのアプリケーション統合コンソールを呼び出すために使用する URL。
SOAP サービス URL	SOAP 要求の送信先 URL。
REST サービス URL	REST 要求の送信先 URL。

## プロパティ

### サマリ

【サマリ】 タブには、BPEL プロセスを実行するエンジンの概要が表示されます。

## ▼ Summary

Date Started:	2018/01/22 10:12 AM
Process Definitions:	132
Cluster Status:	Running
Cluster Monitoring Level:	Normal
Version:	10.3.0 (75971)
Identity Service:	Enabled
Email Service:	⚠ Disabled
Messaging Service:	⚠ Disabled
Cores Detected:	16

次の項目が含まれています。

項目	説明
開始日	エンジンの開始日
プロセス定義	データベースに現在格納されているビジネスプロセスの数（.bpel ファイル）
ステータス（またはクラスタステータス）	<p>プロセスサーバーエンジンのステータスとして指定出来る値は、[実行中] と [停止] です。追加のデータベースメッセージが表示される場合もあります。データベースの詳細情報については、[管理] &gt; [メンテナンス] &gt; [ストレージ] を参照して下さい。</p> <p>表示されるステータスには、Secure Agent クラスタ内のすべてのエンジンが反映されます。例えば、Secure Agent クラスタに 2 つのエンジンがあり、一方が停止している場合、ステータスには [実行中 (1/2 実行中)] と表示されます。</p>
監視レベル（クラスタ監視レベル）	<p>レベルは監視の重大度、つまり警告またはエラーが検出されたことを示します。監視を設定していない場合、またはエンジンが通常どおり実行している場合、レベルは [ノーマル] です。レベルの種類は、[ノーマル]、[警告]、および [エラー] です。</p>
バージョン	プロセスサーバーのバージョン番号
ID サービス 電子メールサービス メッセージサービス	サービスが設定され、有効になっていることを示すステータス。

## サーバーのプロパティ

プロセスサーバーを表示するには、Secure Agent を選択します。

[サーバーの設定] タブでは、エンジンを停止したり再起動せずに設定を変更できます。プロパティを変更して [更新] を選択すると、変更が即座に反映されます。変更はデータベースにも書き込まれるほか、Secure Agent クラスタ内の他のエンジンにもプロパゲートされます。

このトピックの説明に従って、プロセスサーバーの設定を表示および更新してください。一部のプロパティは、Process Developer のシミュレーション設定と同じです。

## プロセス呼び出しでプロキシユーザーを許可

プロセスには、資格情報や匿名ユーザーを持つ通常のプロセスイニシエータとは異なり、指定したユーザーをプロセスイニシエータにできるポリシーアサーション（実行ユーザー名と呼ばれる）を使用できます。必要に応じ、このプロパティを無効にしてポリシーアサーションを無視することもできます。この設定はすべての新しいプロセスインスタンスに適用され、デフォルトでは有効になります。

## Copy/To のターゲットパスの自動作成

BPEL4WS 1.1 仕様に準拠したプロセスにのみ適用されます。WS-BPEL 2.0 のプロセスでは、このプロパティは拡張機能としてプロセス単位で追加できます。詳細については、Process Developer ガイドを参照してください。

このプロパティは、プロセスサーバーが、プロセスインスタンスドキュメントの複雑な変数で、存在しないノードの場所のパスを作成できるかどうかを決定します。割り当てが、存在しないノード（または複数のノード）を参照している場合、標準の BPEL フォールト（bpws:selectionFailure）が BPEL 仕様に従ってスローされます。

このオプションを有効にすると、選択が即座に作成されます。つまり、assign copy TO 操作は、存在しないノードを参照し、それに値を割り当てることができます。このオプションはデフォルトで無効になっています。

## コントリビューションキャッシュ

このキャッシュを設定すると、メモリに保持されるコントリビューションの数を制御できます。このキャッシュにより、データベースを頻繁に読み取る必要がなくなります。一般的に、コントリビューションは大きくありません。ただし、含まれるリソース（カタログエントリ）の数、およびインポートやエクスポートの数に応じてサイズは変わります。

デフォルト値を変更するかどうかは、デプロイされたコントリビューションの数と、使用可能なメモリ量によります。デフォルトのコントリビューションは 100 です。コントリビューションがキャッシュされた場合、データベースでそれをルックアップする必要はありません。100 を超えるコントリビューションがある場合は、この値を増やすことをお勧めします。

このキャッシュ設定には調整があまり必要ありません。アプリケーションサーバーのプロファイリングは、この値の変更がパフォーマンスの向上になるかどうかを判断するために必要になる場合があります。

このキャッシュは、[サーバーのプロパティ] ダイアログでは設定されません。代わりに、エンジンの設定ファイル、aeEngineConfig.xml で設定する必要があります。これは、activvos.war ファイルに含まれています。

## デプロイメントロックのタイムアウト（秒）

クラスタ環境で、あるマシンがデプロイメントロックを取得すると、他のマシンは、デプロイメントが完了し、.bpr デプロイメントが開始するまで待機する必要があります。Secure Agent クラスタ内に多数のマシンがある場合は、各マシンが起動時に多数のシステムの.bprs をデプロイするため、タイムアウト値を大きくすることが必要になる場合があります。デフォルトは 120 秒です。

## デプロイメントプランキャッシュ

デプロイメントプランは、デプロイされたプロセスの各バージョンに対応します（関連する実行中プロセスの処理を含む）。アクティブなプロセスのバージョンは、エンジンのパフォーマンスを向上するためにキャッシュできます。キャッシュされるプランのデフォルト数は 100 です。バージョンの詳細については、「プロセスバージョンのライフサイクル」を参照してください。

## bpws:selectionFailure フォールトを無効化

BPEL4WS 1.1 仕様に準拠したプロセスにのみ適用されます。WS-BPEL 2.0 のプロセスでは、このプロパティは拡張機能としてプロセス単位で追加できます。詳細については、Process Developer ガイドを参照してください。

このオプションにより、割り当て文の FROM 句にある XPath クエリは空のノードセットを返すことができます。その場合、割り当てのターゲットノードは削除されます。

デフォルトでは、このオプションは有効ではありません。クエリ文字列が assign copy FROM から空の選択を返すと、プロセスは bpws:selectionFailure フォールトをスローします。これは、BPEL4WS 仕様で説明されている標準の応答です。

#### 呼び出し作業マネージャスレッドプール (最大)

プロセスサーバーがプロセス呼び出しのために同時に開始できる実行スレッドの最大数を設定します。デフォルトは 25 です。

#### 呼び出し作業マネージャスレッドプール (最小)

プロセスサーバーがプロセス呼び出しに割り当てる実行スレッドの最小数を設定します。デフォルトは 300 です。

#### HTTP 最大接続数/ホストあたりの HTTP 最大接続数

プロセス内にいくつかの HTTP 呼び出しがある場合、この呼び出しによって 100 を超える接続が生成される場合があります。これが発生した場合は、デフォルトの HTTP 最大接続数の値を増やして、いくつものスレッドが HTTP 接続の作成を待ったまま一時停止にならないようにする必要があります（これが起きると、サービス呼び出しのタイムアウトが発生します）。

変更を反映するには、サーバーを再起動する必要があります。

#### メッセージサイズの制限

添付のないマルチパートメッセージについて、ペイロードの最大サイズを設定します。

Informatica Cloud サーバーを使用する場合は、メッセージの最大ペイロードサイズを 5 MB に設定できます。

Secure Agent を使用する場合、デフォルト値は 5 MB ですが、これより大きい値または小さい値に変更できます。

#### メッセージ TTL (秒)

メッセージがプロセスサーバーに到達してからターゲットプロセスにディスパッチされるまでの時間を指定します。デフォルトは 24 時間です。

メッセージの有効期間 (TTL) を設定し、REST または SOAP エンドポイントを使用してプロセスを呼び出す場合、HTTP メッセージがそのメッセージの TTL 期間内にディスパッチされないと、要求は失敗します。次のエラーメッセージで HTTP エラー (503 サービスが使用不可能になりました) が発生します。

定義された TTL を超えたメッセージが破棄されました。

メッセージの TTL は、次のエントリでエンドポイントに適用されます。

- /active-bpel/services/
- /active-bpel/public/rt/
- /active-bpel/rt/
- /active-bpel/public/soap/
- /active-bpel/soap/
- /active-bpel/odata/repository/v4/OdataRepository/Execute
- /active-bpel/tf/
- /activebpel-cert/services/
- /activebpel-cert/rt/
- /activebpel-cert/soap/
- /activebpel-cert/odata/repository/v4/OdataRepository/Execute
- /process-engine/rt/

- /process-engine/public/rt/
- /process-engine/services/
- /process-engine/public/soap/
- /process-engine/soap/

### 添付ファイルのサイズ制限があるメッセージ

添付のあるマルチパートメッセージについて、ペイロードの最大サイズを設定します。

Informatica Cloud サーバーを使用する場合は、メッセージの最大ペイロードサイズを 5 MB に設定できます。添付の最大サイズも 5 MB です。

Secure Agent を使用する場合、デフォルトのメッセージペイロードサイズは 5 MB で、デフォルトの添付サイズは 5 MB です。デフォルトのサイズは変更可能で、これより大きい値または小さい値を設定できます。

### 警告時にも実行中のプロセスを移行

コントリビューションのデプロイメントオプションで [移行] が選択された場合、実行中のプロセスは新しいプロセスバージョンに移行されます。サーバーは、新旧のプロセスプランの互換性を確認し、互換性がないことが判明すると、プロセスサーバーログに警告を生成します。

移行を実行する前に、プロセスサーバーログが少なくとも警告レベルに設定されていることを確認する必要があります。

このプロパティが無効（デフォルト）になっており、1 つ以上の実行中のプロセスインスタンスが警告状態になっている場合は、実行中の各プロセスインスタンスについてコピーが作成され、移行前コピーと新しいコピーが一時停止状態になります。さらに、警告コードと警告メッセージがプロセスサーバーログに追加されます。

このプロパティが有効な場合は、警告状態になっても実行中のインスタンスのコピーは作成されません。警告コードと警告メッセージがプロセスサーバーログに追加されるだけです。

詳細については、警告が生成された場合に実行中のプロセスを移行する方法を参照してください。

### プロセス数

メモリ内の最大プロセス数を指定します。デフォルトは 250 です。0 は制限がないことを示しますが、これは推奨されません。

### プロセスアイドルタイムアウト

呼び出したサービスからの応答を待機している場合など、プロセスのアイドル時間中に、プロセスの状態情報がデータベースに書き込まれるまで待機する秒数を指定します。エンジンのパフォーマンスを向上するには、タイムアウト値を増やします。完全なプロセスの状態を常にデータベース内で保つには、値を減らします。これにより、サーバーの障害時にプロセスのリカバリ時間の発生を回避できます。デフォルトは 10 秒です。

### リソースキャッシュ

保存されたキャッシュ内の WSDL ファイルとその他のリソースの数。デフォルトは 100 です。キャッシュサイズを変更すると、エンジンのパフォーマンスが向上する場合があります。値 1 はキャッシュに制限を設けないことを示しますが、これは推奨されません。

### 実行ごとの実行中のプロセスの中断制限

スケジュールされた 1 回の実行で一時停止する、実行中のプロセスの総数を設定します。制限を設定することで、システム作業マネージャスレッドプールのオーバーロードを制御できます。

例えば、実行中のプロセスの総数が 500 で、制限を 100 に設定するとします。最初のスケジュールされた実行では、実行中の 100 個のプロセスのみが一時停止されます。残りの 400 個は、次にスケジュールされた実行まで待機する必要があります。

デフォルトは-1 です。-1 の値は、実行中のプロセスを一時停止するための制限が設定されていないことを意味し、実行中のプロセスはすべて一時停止されます。

#### 実行中のプロセスの中断タイマー間隔（分）

実行中のプロセスを一時停止する必要がある時間間隔を分単位で設定します。

デフォルトは 1440 分です。これは、24 時間後に、実行中のプロセスの一時停止が始まることを意味します。

#### 画面キャッシュサイズ（MB）

ガイドが使用する画面を保存するためにプロセスサーバーが使用するキャッシュのサイズを設定します。ガイドはパブリッシュされるとサイズが大きくなる場合があります、一般的には、ガイド内のすべてのパスが常に実行されるわけではありません。このキャッシュは、ガイド内で最もよく使用される画面の HTML を保持します。これにより、画面を含むガイドをロードする必要がなくなるため、画面のロードにかかる時間が向上します。

#### 実行時間の長いプロセスを中断するまでの期間（日数）

実行中のプロセスは、解決されるまで [アクティブなプロセス] リストに永続的に残る可能性があります。

実行中のプロセスを保持する日数を入力します（例: 30 日）。実行中のプロセスは、保持期間が到来すると自動的に一時停止します。

デフォルトは 0 です。これは、実行中のプロセスが一時停止されないことを意味します。クラウドサーバーで実行中のプロセスについては、このオプションは Informatica によって決定されます。

#### リカバリの呼び出し時にプロセスを中断

ノードエラーが原因で完了しない呼び出しアクティビティについては、リカバリ時にプロセスを中断できます。プロセスは保留中の呼び出しで中断されます。必要に応じてプロセス例外管理を実行できます。

個々のプロセスは、PDD ファイルのエントリを使用してこの設定をオーバーライドできます。

#### キャッチされないフォールト時にプロセスを中断

WS-BPEL 2.0 仕様によると、キャッチされないフォールトがあるプロセスは終了します。

このオプションを有効にすると、キャッチされないフォールトがあるすべてのプロセスを中断し、それらを中断されたフォールト状態にすることができます。その後、フォールトがあるプロセスに対し、プロセス例外管理（フォールト状態のアクティビティやスコープの再試行や完了など）を実行します。

個々のプロセスは、PDD ファイルのエントリを使用してこの設定をオーバーライドできます。例外管理タイプを参照してください。

プロセス例外管理を参照してください。

#### 実行ごとの中断したプロセスの終了制限

スケジュールされた 1 回の実行で終了する、中断したプロセスの総数を設定します。制限を設定することで、システム作業マネージャスレッドプールのオーバーロードを制御できます。

例えば、中断したプロセスの総数が 500 で、制限を 100 に設定するとします。最初のスケジュールされた実行では、中断された 100 個のプロセスのみが終了されます。残りの 400 個は、次にスケジュールされた実行まで待機する必要があります。

デフォルトは-1 です。-1 の値は、中断したプロセスを終了するための制限が設定されていないことを意味し、中断したプロセスはすべて終了されます。

#### 中断したプロセスの終了タイマー間隔（分）

中断したプロセスを終了する必要がある時間間隔を分単位で設定します。

デフォルトは 1440 分です。これは、24 時間後に、中断したプロセスの終了が始まることを意味します。

## システム作業マネージャスレッドプール（最大）

エンジンが同時に生成できる実行スレッドの最大数を設定します。デフォルトは 300 です。

実行中のスレッド数がこの値と等しい場合、プロセスはフォールトになります。ノードが他のノードに情報をブロードキャストする必要があるときに、使用できるスレッドがないためです。念のため、プロセスサーバーが使用できるセカンダリプールを作成します。これはアプリケーション統合コンソールで行います。プロセスサーバーは、重要なシステム操作を実行する必要がある場合に、このプールのスレッドのみを使用します。

## システム作業マネージャスレッドプール（最小）

エンジンが作業マネージャに割り当てる実行スレッドの最小数を設定します。デフォルトは 25 です。プロセスと、プロセスで実行される可能性のある同時呼び出しを足した数を実行するのに十分な実行スレッドを指定します。

## 一時停止プロセスを次の日数後に終了

一時停止プロセスは、解決されない限り [アクティブなプロセス] リストに永続的に残ります。

一時停止プロセスを保持する日数を入力します（例: 30 日）。一時停止プロセスは、保持期間が到来すると自動的に終了します。

デフォルトは 0 です。これは、中断したプロセスが終了されないことを意味します。クラウドサーバーで実行中のプロセスについては、このオプションは Informatica によって決定されます。

## 一致しない相関受信のタイムアウト

アクティビティがアクティブになる前にメッセージが到着した場合に、プロセスサーバーが相関メッセージをメッセージアクティビティまたはイベントアクティビティと照合するまで待機する秒数を設定します。サーバーで、コンシュームされないメッセージが多数発生することを避けるためにこの値を設定します。

デフォルトは 30 秒です。[一致しない相関受信のタイムアウト] に 0 秒と入力すると、プロセスサーバーは、一致しないすべての相関メッセージを到着時に破棄します。

プロセスサーバーはクラッシュ時、メッセージをリカバリし、関連するコンシューマがメッセージをコンシュームするまで、またはタイムアウトになるまで引き続き待機します。

プロセスサーバーが指定した値を過ぎても待機し続けると、タイムアウトが発生します。タイムアウトが発生すると、プロセスサーバーは相関違反エラーをサーバーログに追加します。この相関違反エラーには、メッセージの詳細は含まれません。相関違反エラーには、AeSecuredLogDatakey と呼ばれる一意のハッシュキー、エンジン ID、プラン ID、キュー ID を連結したもの、および発生時のタイムスタンプ（ミリ秒）が含まれます。

次に示すのは、相関違反エラーのサンプルです。

```
correlationViolation [AeSecuredLogDatakey=786000_123456_1000_1476434734123;tenant=ABCDEF;
planId=123456;partnerLink=MessageEvent1EventPL;operation=Initiate]
```

サービス名、操作名、メッセージパーツなどのメッセージの詳細を表示するには、AeSecuredLogData テーブルで AeSecuredLogDatakey を検索します。

相関違反エラーにはハッシュキーが含まれますが、メッセージの詳細は含まれません。メッセージの詳細はセキュアな情報を含む可能性があるためです。この 2 段階の方法でデータをセキュリティ保護します。

に移動します。

## 入力/出力メッセージをスキーマと照合して検証

サービスの相互作用で使用されるデータを関連するスキーマと照合して検証します。

このオプションを有効にすると、実行を開始する前にデータを検証できます。実行を高速化するにはこのオプションを無効化します。このオプションはデフォルトで有効になっています。

## Web サービス呼び出し/応答のタイムアウト

パフォーマンス上の理由で、受信と一致する応答アクティビティおよび同期呼び出しは、10 分以内に実行しないとタイムアウトになります。タイムアウトエラーが届く場合は、応答または同期呼び出しアクティビティが 10 分以内に実行されないことが原因でプロセスがタイムアウトになるまでの待機時間を長く指定します。

デフォルトは 600 秒です。

## 作業マネージャのアラーム用最大スレッド数

プロセスのアラームでスケジュール設定された作業をディスパッチするため、エンジンが作業マネージャから使用するスレッドの最大数を設定します。100 件台のアラームが同時に発行されると、作業マネージャのすべてのスレッドはアラームをディスパッチするためだけに使用されます。アラームマネージャがすべてのスレッドを使用しているために、パフォーマンスの問題やデッドロックが発生している場合は、この値を増やします。デフォルトは 5 です。

## 作業マネージャのプロセス移行用最大スレッド数

何千個ものプロセスを移行する場合は、スレッド数を増やします。この数は、サーバー上の実行スレッドです。デフォルトは 50 で、新しい受信に割り当てられたスレッド数に影響します。詳細については、上に示したサーバープロパティの「警告時にも実行中のプロセスを移行」を参照してください。

## 作業マネージャのプロセスあたりの最大スレッド数

エンジンが個々のプロセスに対して同時に生成できる実行スレッドの最大数を設定します。デフォルトは 10 です。

## イベントベース接続のスレッドプールプロファイル

AMQP や Kafka などのイベントベースのコネクタを使用する場合、スレッドプールサイズを増やすことで、スレッド数を増やすことができます。

Informatica では、スループットを向上させたい場合にのみスレッドプールサイズを増やすことをお勧めします。これは、追加のスレッドが一部のリソースを占有し、プールが大きくなりすぎることが推奨されないためです。スレッドプールサイズは、すべてのイベントベースの接続に適用されます。

次のプロパティを設定して、スレッドプールのサイズを増やすことができます。

### スレッドプール最小

プールで許可される実行スレッドの最小数を設定します。デフォルトは 10 です。

### スレッドプール最大

プールで許可される実行スレッドの最大数を設定します。デフォルトは 20 です。

### キューサイズ

プロセスサーバーがキューに格納できる非同期アクションの最大数を設定します。デフォルトは 1000 です。

コアスレッドがビジーで、キューがいっぱいになると、プロセスサーバーは最大スレッド数まで追加のスレッドを追加して、追加のタスクを処理します。その後、キューが追加のタスクを受け取っても、プロセスサーバーはそれらのタスクを却下します。

例えば、10 個のタスクをプッシュすると、それらは 10 個のコアスレッドによって処理されます。プロセスサーバーは、追加のタスクをキューに追加します。キューが 1000 に達し、さらにタスクをプッシュしようとする、プロセスサーバーは最大 20 個の追加スレッドを追加します。これは、タスクに割り当てることができるスレッドの最大数です。その後、最大プールサイズに達し、キューがいっぱいになると、プロセスサーバーはタスクを却下します。

**注:** スレッドプール設定を変更したら、プロセスサーバーを再起動して変更を有効にします。

**【サーバーの環境設定】** ページと Secure Agent マシンの aeEngineConfig.xml.mustache ファイルの両方でスレッドプールプロファイル設定を構成した場合、**【サーバーの環境設定】** ページで設定されたプロパティが優先されます。

## JMS 例外リスナ

プロセスサーバーは、管理対象の各接続について JMS 例外リスナを確立します。サーバーは接続の確立後、接続エラーの通知を受信し、指定した期間内に指定した回数の再接続を試みます。デフォルトは 30 秒（30,000 ミリ秒）間に 20 回です。

次の SQL 文を使用してこれらの値を変更します。

```
INSERT INTO AeConfigSettings (ConfigPath, ConfigValue)
VALUES ('MessagingManagers/<manager name>/ReconnectionInterval', '30000');
```

および

```
INSERT INTO AeConfigSettings (ConfigPath, ConfigValue)
VALUES ('MessagingManagers/<manager name>/ReconnectionAttempts', '20');
```

これらの値は、デフォルトの時間が短すぎる場合にのみ変更します。接続リスナに通知されないためにプロセスサーバーが再接続できない場合、これらの設定を変更しても解決しません。

## ログの

**【ロギング】** タブでは、プロセスサーバーおよびプロセスのログレベルを選択できます。

## ▼ Logging

Server Logging Level:	Info ▼
Process Logging Level:	Execution with Service Data ▼
Max Buffer Size:	200
Persist Interval (seconds):	30 
Min Threads	5
Max Threads	150
<input type="checkbox"/> Log all messages	
Logging Base Directory:	../logs

プロパティ名	説明
サーバーログレベル	<p>デフォルトでは、プロセスサーバーはサーバーイベント（サーバープロパティの設定変更、BPR デプロイメント、サーバーの停止と起動、プロセスのエラーなど）のログを生成します。</p> <p>プロセスサーバーには、次のログレベルのいずれかを設定できます。</p> <ul style="list-style-type: none"> <li>- 情報（デフォルト）</li> <li>- エラー</li> <li>- 警告</li> <li>- 重大</li> <li>- 詳細（すべてのレベルを含む）</li> <li>- オフ（ログを生成しない）</li> </ul>
プロセスログレベル	<p>デフォルトでは、プロセスサーバーは実行中のプロセスに関する実行ログを生成します。実行中または完了したプロセスについては、実行ログを表示またはダウンロードできます。実行ログにはアクティビティの実行の開始時間と終了時間が記載されており、フォールトになったプロセスのトラブルシューティングに役立ちます。</p> <p>実行中のプロセスに対して、次のログレベルのいずれかを設定できます。</p> <ul style="list-style-type: none"> <li>- <b>なし</b>: プロセスサーバーは情報をログに記録しません。このオプションを使用すると、エンジンのパフォーマンスが向上します。</li> <li>- <b>実行</b>: これはデフォルトのオプションです。プロセスサーバーは、<i>Will Not Execute</i> 文を除くすべての実行文をログに記録します。<b>【実行】</b>を選択すると、ログファイルのサイズを減らせます。</li> <li>- <b>サービスデータを使用した実行</b>: プロセスサーバーは、すべての実行情報、フォールト情報、およびいくつかの WSIO アクティビティ情報をログに記録します。実行情報については、プロセスサーバーはデッドパスの状態、終了、実行準備完了などのログを記録します。WSIO の場合、プロセスサーバーは、呼び出し、選択、および受信をログに記録します。ただし、データの割り当てや変更に関する情報は記録しません。</li> <li>- <b>データを使用した実行</b>: プロセスサーバーは、すべての実行文（<i>Will Not Execute</i> 文を除くが、変数、式、パートナーリンクデータは含む）をログに記録します。<b>【データを使用した実行】</b>を選択すると、ログファイルのサイズを減らせます。</li> <li>- <b>完全</b>: プロセスサーバーは、デッドパスアクティビティのすべての実行文と <i>Will Not Execute</i> 文をログに記録します。例えば、プロセスサーバーは、実行されないフォールト処理文をすべてログに記録します。</li> </ul> <p><b>【デプロイ済みプロセスバージョン詳細】</b> ページでプロセスログレベルを <b>【システムのデフォルト】</b> に設定することもできます。ログレベルをシステムのデフォルトに設定すると、プロセスバージョンのログレベルは、対応するプロセスサーバーで設定されているプロセスログレベルプロパティに対応します。</p>

プロパティ名	説明
	処理速度を高めるには、ログを実行しないか、最小限に抑えます。Informatica では、ログレベル【なし】または【簡易】の使用を推奨します。
最大バッファサイズ	プロセスサーバーがデータベースに書き込む前にバッファ内に持つ状態の変更、データの変更、およびその他のログイベントの数。 デフォルトは 200 です。
持続間隔 (秒)	プロセスサーバーが、バッファをフラッシュし、ログイベントをメモリからデータベースに書き込む前に待機する秒数。 デフォルトは 30 です。
最小スレッド	プロセスイベントをログに記録するためにプロセスサーバーが起動時に開始するスレッドの最小数。 デフォルトは 5 です。
最大スレッド	プロセスイベントを記録するために任意の時点でアクティブにできる同時実行スレッドの最大数。 デフォルトは 150 です。
すべてのメッセージをログに記録	有効にすると、プロセスサーバーと送受信する OData 要求などの各 WS メッセージのデータ、SOAP エンベロープ、ヘッダー、本文がログに記録されます。すべての HTTP メッセージがファイルシステムにログに記録されます。
ロギングベースディレクトリ	ログファイルのルートディレクトリ。デフォルトのベースディレクトリは、java.io.tmpdir プロパティが参照する場所です。メッセージがログに記録されると、各エンジンは、 <code>&lt;BASE_LOGGING_DIRECTORY&gt;/engine&lt;id&gt;/</code> のルートディレクトリにメッセージログファイルを作成します。  ベースロギングディレクトリの設定には、システムプロパティや環境変数に対して相対的に解決される Ant スタイルのパラメータを含めることができます。例えば、 <code>\${CATALINA_HOME}</code> /ログは、Tomcat サーバーを起動するために使用される CATALINA_HOME 環境変数に対して相対的なログファイルを作成します。

## 監視

【監視】タブで、監視するエンジンのプロパティを選択します。各プロパティには統計やしきい値を指定して、この値に達したときに警告やエラー状態の通知を受けることができます。

次の表に示すように、監視期間の頻度と間隔を指定できます。

フィールド	説明
しきい値期間	統計情報を収集および集計する期間。デフォルト値は 5 分です。
評価頻度	しきい値間隔における統計の評価回数。例えば、しきい値間隔が 5 分で評価頻度が 5 回の場合、統計情報は、5 分間の間、1 分に 1 回収集および集計されます。デフォルト値は 5 回です。

フィールド	説明
最大トラブル項目数	アプリケーション統合コンソールの「監視」ページに表示する、エンジンあたりのエラー/警告項目の数。
警告の監視サービス	この表の後で説明します。

## 警告の監視サービス

監視対象プロパティについてエラーおよび警告が発生した場合に実行するサービスの名前を追加します。このサービスは、MultiSite サイトの使用不可ステータスによって自動的にトリガされた場合も実行されます（*MultiSite* には特別なライセンスが必要です）。

エラーが発生すると、プロセスサーバーは警告サービスをインスタンス化します。次にこのサービスが、監視対象プロパティにエラー状態があることを管理者に通知するなどのアクションを呼び出します。サービスは、エンジンのステータス（実行中または停止状態）も監視できます。

サービスを追加するには、サービス名を入力し、**保存**を選択します。

サービス名は、警告サービスとして使用するため、BPEL プロセスを使用してデプロイされる PDD ファイルで識別される、「マイロール」パートナーリンクサービスです。この名前は、ページを検索することで見つかりません。

サービスの追加後、「詳細の表示」を選択して BPEL プロセスを表示します。

## 監視するサーバープロパティの選択

プロパティを監視するには、次の手順を使用し、**【行の追加】** ボタンを押します。

この行を次のように入力して、プロパティを設定します。

- 監視するプロパティ: 選択リストから項目を選択します。
- レベル: 重要度（エラー、警告、または重大（サーバーを停止））を選択します。
- 統計: 統計を選択します。1 つのプロパティに選択肢が 1 つしかない場合があります。
- 操作: しきい値の関係演算子を選択します。
- しきい値: 評価で使用する、負ではない整数値を追加します。

プロパティの選択後、**【更新】**を選択します。

## プロパティの監視

次のプロパティを監視できます。

### 検出されたクラスタ通信の問題（数）

ノードへのブロードキャストメッセージに関する問題が検出された回数。

### 重大なストレージ例外（数）

ストレージ例外には次が含まれます。

- Select 文の発行に関する問題
- ネットワーク通信エラー
- データベース通信エラー
- データベースユーザーの権限に関する問題

### データベース接続の取得時間（ミリ秒）

データソースからの接続を取得するための待機時間を追跡します。待機時間が長すぎる場合は、接続プールのサイズに問題がある可能性があります。監視には、最大値と平均値があります。

プロセスサーバーのストレージレイヤーは、接続プーリングを実行しません。これは、プール接続へのストレージ実装（通常は `javax.sql.DataSource`）に依存します。接続プールサイズや接続取得時間に関連する低パフォーマンスの問題に対処するには、アプリケーションサーバーまたはデータベースのドキュメントを確認してください。

### デッドロックの再試行（数）

デッドロックの再試行は、エンジンが次を実行しようとしたときに発生します。

- プロセスのロック
- プロセス状態またはジャーナルエントリの書き込み
- ディスパッチについてのアラームの削除
- インバウンド受信の照合
- 内部的なカウンタ値（プロセス ID またはデプロイメント ID）の取得

### 破棄済みの一致しない相関受信（数）

相関プロパティを持つメッセージが、実行中のプロセスインスタンスに転送できず、新しいプロセスインスタンスを作成できない場合、エンジンは設定した時間内にメッセージのディスパッチを繰り返し試みます。エンジンが再試行する、このような一致しないメッセージの数には上限があります。このプロパティは、一致しないメッセージのバッファがいっぱいになったことによって破棄されたメッセージの数を追跡します。

[サーバーのプロパティ] ページに表示される、一致しない相関受信のタイムアウトは、エンジンが、一致しないメッセージをキューに入れておく時間を制御します。この時間に達するとメッセージはプロセスインスタンスに転送されます。タイムアウト値が高すぎると、一致しない受信のプールがいっぱいになる場合があります。ただしこれは、タイムアウト値やバッファサイズではなく、プロセス設計上の問題である可能性もあります。

### クラスタから削除されたエンジン（数）

エンジンがオフラインになり、クラスタブロードキャストメッセージが見つからないことが原因で、エンジンがクラスタから削除された回数。

### プロセスロックの失敗（数）

フェイルオーバーが発生すると、クラスタノードで開始したプロセスインスタンスが、リカバリ中に別のノードに移動されることがあります。稀に、プロセスロックが失敗し、プロセスが一時停止状態（プログラム上）になる場合があります。この問題に対処するため、1 より大きい数（または同等のしきい値）に警告レベルを指定して、このプロパティを設定します。

このプロパティと、プロセスリカバリメッセージのサーバーログを監視することで、プロセスロックによって不要なプロセスの一時停止が発生していないかどうかを確認できます。

### フォールト/一時停止中（フォールト中）プロセス（数）

キャッチされないフォールトが原因で、フォールト状態または一時停止状態で終了しているプロセスの数。

このエンジンで実行中のプロセスが失敗した場合には、通知を受けることができます。

### プランキャッシュの効率（パーセント）

デプロイメントのキャッシュ設定が正しいかどうかを確認できます。

デプロイメントプランは、デプロイされたプロセスの各バージョンに対応します（関連する実行中プロセスの処理を含む）。アクティブなプロセスのバージョンは、エンジンのパフォーマンスを向上するためにキャッシュできます。キャッシュされるプランのデフォルト数は 100 です。

### プランキャッシュの削除（数）

プロセスのプランがキャッシュから削除された回数。

### プランキャッシュのターンオーバー（パーセント）

新しいプランのロードで、古いプランがキャッシュから強制的に削除される頻度。

### プロセスキャッシュの効率（パーセント）

エンジン設定には、ストレージに強制移動されるまではメモリに保持できる、プロセスの最大数のカウントも含まれます。

プロセスは、ロジックを実行している場合、または休止状態だが別のメッセージ、アラーム、呼び出しへの応答を待機している場合、メモリにキャッシュされます。このプロパティは、メモリから読み取られるプロセスと、ストレージから読み取られるプロセスインスタンスの割合を報告します。例えば、100%はすべてのプロセスがメモリキャッシュから読み取られることを示します。

「サーバーのプロパティ」ページでは、「プロセス数」と「プロセスアイドルタイムアウト」の値を設定できます。「プロセス数」設定は、キャッシュのサイズを制御します。「プロセスアイドルタイムアウト」設定は、アイドル状態のプロセスをキャッシュ内に維持する期間を制御します。プロセスキャッシュ効率のパーセンテージが低く、プロセスに双方向呼び出しが含まれているか、プロセスが複数のインバウンドメッセージを処理できる場合は、「プロセス数」と「プロセスアイドルタイムアウト」を増やすことをお勧めします。これは、プロセスをメモリに維持するのに役立ちます。ただし、プロセスの実行時間が長く、メッセージを定期的にしか受信しない場合、プロセスキャッシュ効率のパーセンテージを低くしても問題はありません。

### プロセス数の超過（数）

「サーバーのプロパティ」には、メモリ内の最大プロセス数を指定する「プロセス数」オプションがあります。プロセス数が「プロセス数」に設定した値を超えた場合、このプロパティに基づいた警告を作成できます。

例えば、「プロセス数」が50に設定されており、プロセス数が55に達したら警告を受け取るようにするには、しきい値が5以上になったときに警告が発行されるようにこの値を設定します。

### プランの取得時間（数）

プロセスプランのメモリへのロードにかかる時間。

### プロセスの取得時間（ミリ秒）

プロセスの取得時間は、負荷がかかる状況でこの操作が大幅に高負荷に向かっているかどうかを判断するのに便利です。監視には、最大値と平均値があります。このプロパティには、プロセスに対するロックを取得する時間と、ストレージからの状態のリストア（必要な場合）が含まれます。

このプロパティは、プロセスキャッシュの効率と組み合わせて動作します。

### XSL 変換の実行時間（ミリ秒）

`doXslTransform()` カスタム関数内での変換を実行するのにかかる時間。

### ID プロバイダの照会時間（ミリ秒）

ID プロバイダの照会にかかるミリ秒単位の時間。例えば、あるユーザーのグループをリストするためのLDAPサーバーへの照会要求に20ミリ秒かかる場合、その秒数が、そのインスタンスに対するこのメトリックの値です。

このプロパティを使用すると、照会時間の急上昇を追跡する際に、LDAPプロバイダの速度遅延の問題を特定できます。

### プロセスの保存時間（ミリ秒）

プロセス状態と変数をデータベースに保存するために必要なミリ秒数。

しきい値は、プロセスの構成、変数の数、可変データのサイズに大きく左右されます。このプロパティは、永続性の設定が「完全」または「永続」のプロセスに対してのみ有効です。

### メッセージの検証時間（ミリ秒）

受信、呼び出し、その他のアクティビティからの入出力メッセージをエンジンが検証する時間を報告します。

この検証は、[入力/出力メッセージをスキーマと照合して検証] という設定ページで有効にします。有効な場合、すべてのメッセージが検証されます。検証は、個々のパートナーリンクでも、ポリシーアサーションを介して有効または無効にできます。

このプロパティは、BPEL 検証アクティビティ、または assign アクティビティのオプション検証属性を介した、明示的な変数検証にかかる時間は追跡しません。

すべてのメッセージまたは選択したメッセージの検証を無効にすると、処理を高速化できます。メッセージの検証に時間がかかりすぎる場合は、いくつかの手順に分けて実行できます。まず、特定の種類のメッセージを細かく管理できる、追加したパートナーリンクの [メッセージ検証] ポリシーアサーションでプロセスを再デプロイします。すべてのプロセスに対するメッセージ検証の無効化は、[サーバープロパティ] を無効にすることでも実現できます。

### 作業マネージャの作業開始の遅延（ミリ秒）

作業項目要求のスケジュールから実際の作業開始までにかかる時間は、作業マネージャプールを調整するのに役立ちます。


遅延が長くなると、作業量の処理に使用できるスレッドが十分でなくなる場合があります。監視には、最大値と平均値があります。

デフォルトの作業マネージャを使用している場合、作業プールのサイズは [サーバーのプロパティ] ページで設定できます。アプリケーションサーバーが提供する作業マネージャ実装を使用している場合、プールのサイズとスレッドの優先順位はアプリケーションサーバーの管理コンソールで設定できるはずです。

## 受信キュー

このページのコントロールを使用すると、キューに入れられた受信をいくつメモリに保持するかを決めるサイズを設定し、データベースへのクエリの日数を減らすことができます。サイズ設定は、サーバーメモリ制限に応じて変わります。

### ▼ Receive Queue

Unmatched Correlated Receive Timeout (seconds):	<input type="text" value="30"/>	
Unmatched Correlated Receive Count:	<input type="text" value="50"/>	
Queued Receive Cache Size:	<input type="text" value="5000"/>	

このページの情報は次のとおりです。

### 一致しない相関受信のタイムアウト

相関メッセージが受信アクティビティに一致するのをエンジンが待機する時間（秒）。これは、アクティビティがアクティブになる前にメッセージが到着した場合に限ります。相関メッセージの一致にかかる時間が指定した時間を超えると、エンジンはメッセージを破棄し、相関違反の例外が示されます。

デフォルト値は 30 秒です。サーバーで、コンシュームされないメッセージが多数発生することを避けるため、この値には低い数字を設定します。この値をゼロ秒に指定するように設定すると、エンジンは一致しないすべての関連メッセージを即座に破棄します。

**時計アイコン**をクリックすると、期間を選択するためのダイアログボックスが表示されます。

詳細については、「サーバーのプロパティ」を参照してください。

#### 一致しない関連受信の数

エンジンが一度にメモリに保持する、一致しない関連受信の最大数。この制限値を超えると、エンジンは一致しない関連受信を古いものから順に却下し、新しい受信のために空きを作ります。

#### キューに入れられた受信のキャッシュサイズ

システムが他の要求でビジー状態のときにアクティビティ情報を保持するキャッシュメモリのサイズ。

これらの 3 つのフィールドに表示される値は変更することもできます。

イベントが表示されたら、受信を選択し、詳細を表示するパートナーリンクを選択します。ウィンドウが表示され、受信アクティビティが実行される BPEL プロセスの場所を確認できます。この受信アクティビティと関連付けられた、関連プロパティのエイリアスとデータ（ある場合）を確認することもできます。

## ディスパッチサービス

ディスパッチサービスを使用すると、プロセスサーバーに対す要求を調整するディスパッチ設定を作成できます。

要求とは、HTTP/S メッセージ、メッセージイベント、Transport Layer Security (TLS) などの要求です。要求をキューに格納したり、要求をバッチ単位でディスパッチしたり、同時実行する要求の最大数を制御することができます。

例えば、非同期要求をプロセスサーバーにすばやくディスパッチするプロセスをよく使用するとします。実行時に、このような要求を何千件も同時に実行すると、サーバーや呼び出したサービスが過負荷に陥る可能性があります。ディスパッチサービスを使用すると、要求をバッチ単位で送信して、限られた数の要求のみが同時に実行されるようにすることができます。

Secure Agent で実行するサービスに対してディスパッチサービスを使用することもできます。クラウドサーバーで実行するサービスに対しては、ディスパッチサービスは使用できません。

サービスまたはプロセスグループに対してディスパッチ設定を作成します。実行時に、アプリケーションの統合によって、次の優先順位でディスパッチ設定が選択されます。

1. サービス名
2. プロセスグループ
3. システムのデフォルト

例えば、processA\_event1 というサービスに、次のディスパッチ設定が適用対象としてあるとします。

- processA\_event1 (サービス名)
- GroupABC (プロセスグループ)
- SystemDefault

プロセスサーバーが processA\_event1 を実行すると、processA\_event1 ディスパッチ設定が適用されます。

ディスパッチサービスは、境界イベントを含むサブプロセス呼び出しを除く、すべての要求に適用されます。例えば、サブプロセスステップにメッセージイベントが含まれている場合、ディスパッチサービスはこのサブプロセスには適用されません。

ただし、サービス呼び出しステップを使用してサブプロセスを追加する場合、ディスパッチは、境界イベントのあるサブプロセスを含むすべてのサブプロセス設定に適用されます。

## デフォルトおよびカスタムのディスパッチサービス

プロセスサーバーにはデフォルトのディスパッチサービスがあります。

次の表は、デフォルトのディスパッチサービスとその目的を示しています。

デフォルトのディスパッチサービス	目的
SystemDefault	すべてのサービスに適用されます。
avBusinessConnectionRESTService	サービスコネクタに適用されます。
avCreateAnyEntityService	データベース接続を使用した、ホストランタイムによるレコードの PUT 操作を有効にします。
avDeleteAnyEntityService	データベース接続を使用した、ホストランタイムによるレコードの DELETE 操作を有効にします。
avHostEnvironmentRuntimeAccess	データベース接続を使用した、ホストランタイムによるレコードへのアクセスを有効にします。
avHostRuntimeCall	データベース接続を使用した、ホストランタイムによるレコードの GET 操作を有効にします。
avProxyToProcess	プロセスによる他のプロセスのサービスとしての呼び出しを有効にします。
jmsEnqueueService	JMS 自動化ステップに適用されます。
sfHostEnvironmentRuntimeService	Salesforce 接続を使用した、ホストランタイムによる Salesforce データベースへのアクセスを有効にします。

カスタムのディスパッチ設定を作成するには、**【サーバーの環境設定】**に移動し、Secure Agent を選択します。次に、**【キュー】 > 【ディスパッチキュー】**に移動します。

## ディスパッチサービスのコンポーネント

ディスパッチサービスは同時進行プールとメモリ内キューおよびパーシステントキューを使用して、プロセスサーバーへの要求の流れを制御します。

ディスパッチサービスは、次のプールとキューを使用します。

### 同時進行プール

同時進行プールは、プロセスサーバーが一度に処理する要求を保持します。要求を処理するために使用できるリスナの数、最大同時進行の値によって異なります。

### メモリ内キュー

メモリ内キューは、ディスパッチサービスのメモリ内に要求を保持します。最大同時進行の値に達すると、ディスパッチサービスは要求をメモリ内キューに移します。

同時進行プールに領域がある場合は、要求はメモリ内キューから同時進行プールに移されます。

パーシステントを有効にしていない限り、Secure Agent が再起動すると、メモリ内キューの要求は失われます。

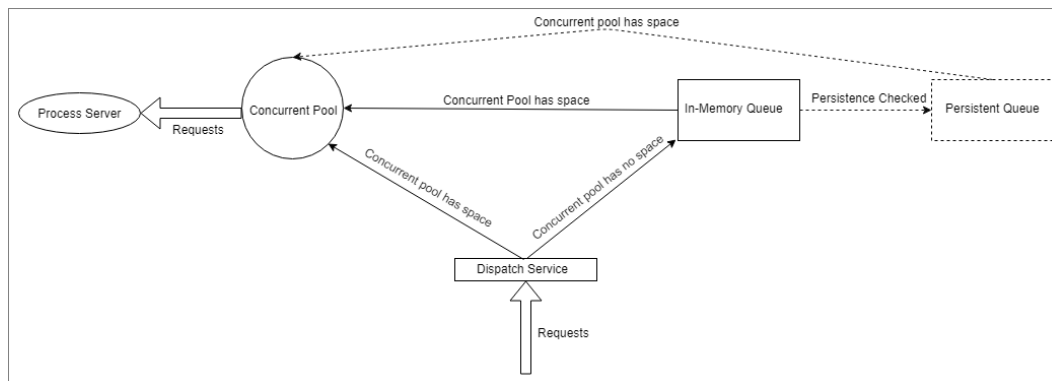
### パーシステントキュー

パーシステントを有効にしている場合、パーシステントキューではメモリ内キューの要求を保持します。パーシステントキューは、データベース内に要求を保持します。

リスナが解放されると、要求はパーシステントキューから同時進行プールに移されます。

Secure Agent が再起動し、要求がパーシステントキューではなくメモリ内キューにのみ存在する場合、要求は失われます。

次の図は、ディスパッチサービスが要求を調整する方法を示しています。



詳細については、『[Sample Dispatch Configurations \(ページ 29\)](#)』を参照してください。

## カスタムディスパッチ設定の作成

カスタムディスパッチ設定を作成するには、次の手順を実行します。

1. アプリケーション統合コンソールにログインします。
2. **【サーバーの環境設定】** に移動し、**【Secure Agent】** を選択します。
3. **【キュー】** > **【ディスパッチキュー】** > **【追加】** に移動します。
4. **【新しいディスパッチキュー】** ダイアログボックスでプロパティを設定し、**【OK】** をクリックします。

次の図は、00000S サービスの**【新しいディスパッチキュー】** ダイアログボックスの例を示しています。

New Dispatch Queue

↗ ⊗

At runtime, the dispatch configuration used for a particular request is chosen based on matching the configuration name in the following order of precedence: Service Name, Process Group, Tenant, System Default

Name

00000S

Max Concurrent

15

Max In-Memory

3

Max Queued

2

Timeout (seconds)

240

Persistent

☐

?

OK

Cancel

## ディスパッチサービスの設定

カスタムディスパッチ設定を作成するには、次のプロパティを設定します。

プロパティ	説明																					
名前	<p>サービス定義の名前。</p> <p>サービス定義名を表示するには、<b>【プロセス】 &gt; 【プロセスバージョン番号の選択】 &gt; 【マイロール】 &gt; 【サービス列】</b> にアクセスします。</p> <p>例えば、<b>【デプロイされたプロセスバージョンの詳細】</b> ページの <b>【マイロール】</b> セクションでは、サービス名は 0BQGwyKeJVbie6UGrJHaqI/Print_Hello_World-1 です。</p> <table><tr><th colspan="7">My Role</th></tr><tr><th>Partner Link</th><th>Type</th><th>Allowed Roles</th><th>Role</th><th>Binding</th><th>Service</th><th>Policy</th></tr><tr><td>ProcessConsumer</td><td>spPartnerLinkType</td><td></td><td>Consumer</td><td>MSG</td><td>0BQGwyKaJVbie6UGrJHaqI/Print_Hello_World-1</td><td>view</td></tr></table> <p>サービス定義名を入力すると、ディスパッチ設定はそのサービスを含むプロセスに関係なく、サービスのすべてのインスタンスに適用されます。</p> <p>保存したディスパッチ設定の名前を変更することはできません。名前を編集するには、ディスパッチ設定を削除して、新しいディスパッチ設定を作成します。</p>	My Role							Partner Link	Type	Allowed Roles	Role	Binding	Service	Policy	ProcessConsumer	spPartnerLinkType		Consumer	MSG	0BQGwyKaJVbie6UGrJHaqI/Print_Hello_World-1	view
My Role																						
Partner Link	Type	Allowed Roles	Role	Binding	Service	Policy																
ProcessConsumer	spPartnerLinkType		Consumer	MSG	0BQGwyKaJVbie6UGrJHaqI/Print_Hello_World-1	view																
最大同時進行	<p>プロセスサーバーが同時進行プールに保持できる要求の最大数。</p> <p>デフォルト: 15</p>																					
最大メモリ内	<p>プロセスサーバーがメモリ内キューに保持できる要求の最大数。</p> <p>パーシステントを有効にしている場合、ディスパッチサービスは <b>【最大キュー格納】</b> 値を超える要求を却下します。</p> <p>パーシステントを有効にしていない場合、ディスパッチサービスは <b>【最大インメモリ】</b> 値を超える要求を却下します。</p> <p>デフォルト: 300</p>																					

プロパティ	説明
最大キュー	<p>パースistentキュー内の要求の最大数。</p> <p>パースistentを有効にしている場合、ディスパッチキューサービスは【最大キュー格納】値を超える要求を却下します。</p> <p>パースistentを有効にしていない場合、ディスパッチサービスは【最大インメモリ】値を超える要求を却下します。</p> <p>デフォルト: 300</p>
タイムアウト (秒)	<p>ディスパッチサービス内で1つの要求の実行にかけることができる時間の最大値 (秒)。</p> <p>要求の実行にかかる時間がこのタイムアウト値より長くなると、その要求はディスパッチサービスを離れ、次の保留中の要求が実行されます。</p> <p>要求がディスパッチサービスのタイムアウトになると、プロセスサーバーはディスパッチ設定外で引き続き実行されます。その結果、【プロセス】リストと【サーバーの環境設定】 &gt; 【キュー】 &gt; 【ディスパッチキュー】 リストで異なる値が表示される場合があります。</p> <p>デフォルト: 300 秒。</p>
パースistent	<p>保留中の要求をパースistentキューに保存するには、【パースistent】オプションを選択します。このオプションを選択し、Secure Agent が再起動した場合、ディスパッチ設定はキューに入っている要求を引き続き実行します。</p> <p>【パースistent】オプションを選択していない場合、保留中の要求はメモリ内キューにのみ保存され、Secure Agent が再起動した場合は失われます。</p> <p>デフォルト: 未選択。</p>

これらのプロパティを使用して要求を調整する方法については、[Sample Dispatch Configurations \(ページ 29\)](#)を参照してください。

## ディスパッチ設定の監視

ディスパッチ設定を監視するには、【ディスパッチキュー】 ページを使用します。

次の表は、【ディスパッチキュー】 ページのカラムを示しています。

カラム	説明
名前	ディスパッチ設定名。
実行中	ディスパッチ設定に基づいて実行されている要求の数。
キューに格納	<p>ディスパッチ設定に基づいてキューに格納されている要求の数。</p> <p>要求は、次のいずれかのイベントが発生すると、【キューに格納】 から【実行中】 に移動されます。</p> <ul style="list-style-type: none"> <li>- 要求が完了した。</li> <li>- 要求がタイムアウトになった。</li> <li>- <a href="#">purged queued requests</a> ために要求がキューから移動された。</li> </ul>
平均時間 (ミリ秒)	同時実行プールのリスナが要求に対して動作するミリ秒単位の平均時間。
コンシューム済み	ディスパッチ設定を通過した要求の数。この数には、却下されたメッセージは含まれません。

カラム	説明
却下	ディスパッチ設定で却下された要求の数。 この数には、タイムアウトした要求は含まれません。詳細については、 <a href="#">Timeout (Seconds) (ページ 28)</a> を参照してください。
ステータス	ディスパッチ設定のステータス。 ディスパッチ設定は、次の状態になります。 <ul style="list-style-type: none"> <li>- <b>アクティブ</b>: ディスパッチ設定は実行中で、キュー内の要求をプロセスサーバーに送信できます。</li> <li>- <b>一時停止中</b>: ディスパッチ設定は一時停止しています。保留中のキュー内の要求は、ディスパッチ設定が再開されるまでキューに残ります。</li> <li>- <b>保留中の削除</b>: コンSUMする必要がある要求がキュー内にあります。このステータスを見つけたら、ディスパッチ設定を <a href="#">delete</a> し、ブラウザを更新します。</li> </ul>

次の表は、ディスパッチ設定に対して実行できるアクションを示しています。

アクション	説明
統計のリセット	<b>【実行中】</b> と <b>【平均時間（ミリ秒）】</b> の値をゼロにします。
実行を中断	プロセスサーバーへの要求の送信を停止します。要求は、ディスパッチ設定を再開するまでキュー内に残ります。
実行の再開	エンジンへの新しい要求のディスパッチを再開します。
キューに格納された要求のページ	一時停止中のすべての要求をメモリ内キューと永続キューから削除します。プロセスサーバーは、これらの要求を実行しません。 このアクションは、通常とは異なる、キューのクリアが必要となる状況で使用します。
構成の削除	ディスパッチ設定を削除します。ブラウザを更新するまで、ステータスは <b>【保留中の削除】</b> となります。 <b>【構成の削除】</b> を選択すると、それ以降要求はプロセスサーバーに送られません。キュー内に要求がある場合、設定は、すべての要求がCONSUMされた後で削除されます。

## ディスパッチ設定の例

このセクションでは、ディスパッチ設定の例をいくつか説明します。

### 例 1

以下の表にディスパッチ設定の例を示します。

プロパティ	値
最大同時進行	15
最大メモリ内	200

プロパティ	値
最大キュー	1000
タイムアウト（秒）	55

ディスパッチ設定は 10,000 件の要求を受信します。

次の表は、各要求に 50 秒または 60 秒かかり、パースistentが有効になっている場合、または有効になっていない場合に発生するイベントを示しています。

パースistent/要求あたりにかかる時間	50	60
有効になっていない	<ul style="list-style-type: none"> <li>- 50 秒ごとに、15 件の要求が【キュー】列から【実行中】列に移動します。</li> <li>- ディスパッチサービスは 200 件の要求を処理し、9800 件の要求を拒否します。プロセスサーバーは拒否されたメッセージを実行しません。</li> <li>- 200 件の要求はすべて正常に処理されます。これは<b>タイムアウト</b>値が要求の完了にかかる時間よりも長いからです。</li> <li>- [Consumed (処理済み)] 列は 200 で、[拒否] 列は 9800 です。</li> <li>- 200 件の要求を処理する前に Secure Agent が再起動すると、キューに入った要求はパースistentキューではなくメモリ内キューに保持されているため失われます。</li> </ul>	<ul style="list-style-type: none"> <li>- 55 秒ごとに、15 件の要求が【キュー】列から【実行中】列に移動します。これは、<b>タイムアウト</b>値が要求の完了にかかる時間よりも短いからです。</li> <li>- ディスパッチサービスは 200 件の要求を処理し、9800 件の要求を拒否します。プロセスサーバーは拒否されたメッセージを実行しません。</li> <li>- <b>タイムアウト</b>値が要求の完了にかかる時間より短いため、200 件の要求はディスパッチサービスの範囲内では完了しません。プロセスサーバーはタイムアウトメッセージがディスパッチサービスを離れた後にそれらのメッセージを引き続き実行します。</li> <li>- [Consumed (処理済み)] 列は 200 で、[拒否] 列は 9800 です。</li> <li>- 200 件の要求が処理される前に Secure Agent が再起動すると、キューに入った要求は失われます。</li> </ul>
有効	<ul style="list-style-type: none"> <li>- 50 秒ごとに、15 件の要求が【キュー】列から【実行中】列に移動します。</li> <li>- ディスパッチサービスは 1000 件の要求を処理し、9000 件の要求を拒否します。プロセスサーバーは拒否された要求を実行しません。</li> <li>- ディスパッチサービスは 1000 件の要求を正常に処理します。これは<b>タイムアウト</b>値が要求の完了にかかる時間よりも長いからです。</li> <li>- [Consumed (処理済み)] 列は 1000 で、[拒否] 列は 9000 です。</li> <li>- 1000 件の要求が処理される前に Secure Agent が再起動すると、保留中の要求は再起動の後に実行されます。これは、要求がパースistentキューに保存されていたためです。</li> </ul>	<ul style="list-style-type: none"> <li>- 55 秒ごとに、15 件の要求が【キュー】列から【実行中】列に移動します。これは、<b>タイムアウト</b>値が要求の完了にかかる時間よりも短いからです。</li> <li>- ディスパッチサービスは 1000 件の要求を処理し、9000 件の要求を拒否します。プロセスサーバーは拒否された要求を実行しません。</li> <li>- <b>タイムアウト</b>値が要求の完了にかかる時間より短いため、1000 件の要求はディスパッチサービスの範囲内では完了しません。プロセスサーバーはタイムアウトメッセージがディスパッチサービスを離れた後にそれらのメッセージを引き続き実行します。</li> <li>- [Consumed (処理済み)] 列は 1000 で、[拒否] 列は 9000 です。</li> <li>- [Consumed (処理済み)] 列は 200 で、[拒否] 列は 9800 です。</li> <li>- 1000 件の要求が処理される前に Secure Agent が再起動すると、保留中の要求は再起動の後に実行されます。これは、要求がパースistentキューに保存されていたためです。</li> </ul>

## 例 2

以下の表にディスパッチ設定の例を示します。

プロパティ	値
最大同時進行	4
最大メモリ内	20
最大キュー	10
タイムアウト（秒）	7

ディスパッチ設定は 50 件の要求を受信します。

次の表は、各要求に 50 秒または 60 秒かかり、パースistentが有効になっている場合、または有効になっていない場合に発生するイベントを示しています。

パースistent/要求あたりにかかる時間	5	10
有効になっていない	<ul style="list-style-type: none"> <li>- 5 秒ごとに、4 件の要求が <b>【キュー】</b> 列から <b>【実行中】</b> 列に移動します。</li> <li>- ディスパッチサービスは 20 件の要求を処理し、30 件の要求を拒否します。プロセスサーバーは拒否された要求を実行しません。</li> <li>- ディスパッチサービスは 20 件の要求を正常に処理します。これは<b>タイムアウト</b>値が要求の完了にかかる時間よりも長いからです。</li> <li>- <b>【Consumed (処理済み)】</b> 列は 20 で、<b>【拒否】</b> 列は 30 です。</li> <li>- 20 件の要求を処理する前に Secure Agent が再起動すると、キューに入った要求はパースistentキューではなくメモリ内キューに保持されているため失われます。</li> </ul>	<ul style="list-style-type: none"> <li>- 7 秒ごとに、4 件の要求が <b>【キュー】</b> 列から <b>【実行中】</b> 列に移動します。</li> <li>- ディスパッチサービスは 20 件の要求を処理し、30 件の要求を拒否します。プロセスサーバーは拒否された要求を実行しません。</li> <li>- <b>タイムアウト</b>値が要求の完了にかかる時間より短いため、20 件の要求はディスパッチサービスの範囲内では完了しません。プロセスサーバーはタイムアウトメッセージがディスパッチサービスを離れた後にそれらのメッセージを引き続き実行します。</li> <li>- <b>【Consumed (処理済み)】</b> 列は 20 で、<b>【拒否】</b> 列は 30 です。</li> <li>- 20 件の要求を処理する前に Secure Agent が再起動すると、キューに入った要求はパースistentキューではなくメモリ内キューに保持されているため失われます。</li> </ul>
有効	<ul style="list-style-type: none"> <li>- 5 秒ごとに、4 件の要求が <b>【キュー】</b> 列から <b>【実行中】</b> 列に移動します。</li> <li>- ディスパッチサービスは 20 件の要求を処理し、30 件の要求を拒否します。プロセスサーバーは拒否された要求を実行しません。</li> <li>- ディスパッチサービスは 20 件の要求を正常に処理します。これは<b>タイムアウト</b>値が要求の完了にかかる時間よりも長いからです。</li> <li>- <b>【Consumed (処理済み)】</b> 列は 20 で、<b>【拒否】</b> 列は 30 です。</li> <li>- 30 件の要求が処理される前に Secure Agent が再起動すると、保留中の要求は再起動の後に実行されます。これは、要求がパースistentキューに保存されていたからです。</li> </ul>	<ul style="list-style-type: none"> <li>- 7 秒ごとに、4 件の要求が <b>【キュー】</b> 列から <b>【実行中】</b> 列に移動します。</li> <li>- ディスパッチサービスは 20 件の要求を処理し、30 件の要求を拒否します。プロセスサーバーは拒否された要求を実行しません。</li> <li>- <b>タイムアウト</b>値が要求の完了にかかる時間より短いため、20 件の要求はディスパッチサービスの範囲内では完了しません。プロセスサーバーはタイムアウト要求がディスパッチサービスを離れた後にそれらの要求を引き続き実行します。</li> <li>- <b>【Consumed (処理済み)】</b> 列は 20 で、<b>【拒否】</b> 列は 30 です。</li> <li>- 30 件の要求が処理される前に Secure Agent が再起動すると、保留中の要求は再起動の後に実行されます。これは、要求がパースistentキューに保存されていたからです。</li> </ul>

## ディスパッチサービスのトラブルシューティング

ディスパッチサービスを使用する場合、先に進めなくなる状況に遭遇する場合があります。

**【ディスパッチサービスの要求】** ページの **【実行中】** 列の要求の数は、**【アクティブなプロセス】** ページの実行中のプロセスの数と異なります。要求がタイムアウトしていないことは確実です。

考えられる理由: サブプロセスを呼び出すプロセスを呼び出しました。

考えられる解決策: **【デプロイされたプロセスバージョンの詳細】** ページで、プロセスとサブプロセスの **【永続タイプ】** 設定を変更し、プロセスの呼び出しを再試行します。

パースistent設定（[Full（完全）] または [永続的]）と非パースistent設定（[Final（最終）]、[Brief（概要）]、または [なし]）を切り替えます。

プロセスを5回呼び出しましたが、[キュー] 列には何百件もの要求があります。これらの要求が移動する速度が遅く、Secure Agent がクラッシュするのではないかと思います。

考えられる理由: サブプロセスを呼び出してさらにサブプロセス呼び出すプロセスを呼び出しました。この階層が深くなる可能性があります。

考えられる解決策: プロセスが呼び出すサブプロセスに対して複数のディスパッチサービスを作成します。それぞれのディスパッチ設定はサービスを個別に実行して調整し、その結果サービスの実行が高速になります。

## ディスパッチサービスに関連するその他のコントロール

ディスパッチサービスを監視する際に、プロセスサーバーへの要求のオーバーフローを監視して対処するためにその他の構成設定を使用できます。問題があると思われる場合、ディスパッチサービスの最大キュー値を超えていることが原因で起こっている可能性があります。

次のコマンドを使用して、要求が拒否されたかどうかを判断します。

- [サーバーの環境設定] > [プロパティ] > [サーバー設定] タブで、[Unmatched Correlated Receive Timeout（相関受信不一致タイムアウト）] の値を設定できます。詳細については、「サーバーのプロパティ」を参照してください。
- [サーバーの環境設定] > [監視] タブで、破棄された不一致の相関受信のプロパティを追加できます。詳細については、「監視」を参照してください。
- 拒否された要求に関連するエラーがあるかどうか、プロセスサーバーログをチェックします。詳細については、「サーバーログ」を参照してください。

## ストレージ

**注:** このコマンドは、エージェントを選択した場合にのみ表示されます。

プロセスサーバーには、インストール時に設定したデータベース設定に基づいた永続ストレージがあります。データベースは、エンジンを実行する前に設定する必要があります。

[ストレージ] ページにはデータベース設定プロパティが表示され、ここでデータベースを管理できます。

Properties	Monitors	Queues	Storage	System Services
JNDI Location:		java:comp/env/jdbc/ProcessEngine		
Database Type:		postgre		
<button>Clean Up Storage ...</button>				

表示されるプロパティは次のとおりです。

プロパティ	説明
JNDI の場所	データベースを参照する場所を指定する Java Naming and Directory Interface (JNDI) のコンテキスト。例えば、jdbc/ActiveVOS です。
データベースタイプ	mysql などのタイプ。サポート対象タイプのリストについては、このヘルプにある、プロセスサーバーのインストール、設定、およびデプロイに関する説明で、前提条件に関するトピックを参照してください。
ユーザー名	プロセスサーバーがデータベースにアクセスするためのユーザー名（必要な場合）。

## スケジュールされたメンテナンス

[ストレージ] タブを使用すると、項目をデータベースから削除するタイミングと頻度を指定できます。

スケジュールの削除	
頻度	<ul style="list-style-type: none"><li>- <b>【なし】</b> は、スケジュールされたメンテナンスを行わないようにエンジンをリセットします。</li><li>- <b>【日中】</b>、<b>【毎日】</b>、<b>【毎週】</b>、または <b>【毎月】</b> は、スケジュールの頻度を設定します。</li></ul>
間隔	<ul style="list-style-type: none"><li>- <b>【日中】</b> の場合は、分または時間と数字を選択します。例えば、90 分を選択します。</li><li>- <b>【毎週】</b> の場合、1 日以上の日曜日を選択します。例えば、<b>【月曜日】</b> を選択すると、毎週月曜日にデータベース項目を削除できます。毎週月曜日と木曜日に削除するには、<b>【月曜日】</b> と <b>【木曜日】</b> を選択します。</li><li>- <b>【毎月】</b> の場合は、月の 1 番目、2 番目、3 番目、4 番目、または最後の日を選択します。例えば、月の最後の月曜日に項目を削除するには、<b>【最終月曜日】</b> を選択します。</li></ul>
開始時刻 (GMT)	時刻を入力するか、 <b>【日付選択ツール】</b> を選択して時刻を入力します。 <b>【日付選択ツール】</b> で、 <b>【今】</b> を選択して現在の時刻を入力します。
オフラインコントロールビューションの削除	古いプロセスバージョンを削除するには、この設定を選択します。
完了したプロセスの削除 フォールトになったプロセスの削除	正常に完了した ( <b>【完了】</b> ) または正常に完了しなかった ( <b>【フォールト】</b> ) 古いプロセスを削除するには、この設定を選択します。

スケジュールの削除	
デフォルトのフォールトプロセス保持時間	<p>プロセスサーバーがフォールトになったプロセスをデータベースに保持する日数、時間数、または分数を入力します。デフォルトは 72 時間（3 日間）です。</p> <p>注: <b>【デプロイされたプロセスバージョンの詳細】</b> ページで <b>【プロセスインスタンス保持時間】</b> 値を指定できます。この値はプロセスに固有で、一般的な <b>【デフォルトのフォールトプロセス保持時間】</b> 値より優先されます。</p> <p><b>【プロセスインスタンス保持時間】</b> 値の詳細については、監視マニュアルのカタログとリソースの管理のセクションにある、デプロイされたプロセスバージョンの詳細を参照してください。</p> <p>フォールトになったプロセスについて電子メールの警告を受け取るように電子メールサービスを設定した場合は、フォールトになったプロセスごとに 1 時間あたり最大 1 通の電子メールを受信します。</p> <p>例えば、5 秒ごとにフォールトになる、プロセス A とプロセス B という 2 つのプロセスがある場合、1 時間以内に受信するのはプロセス A に関するメールとプロセス B に関するメールの 2 通です。</p>
次より古いサーバーログエントリを削除	<p>プロセスサーバーがサーバーログデータを保持する日数、時間数、または分数を入力します。</p> <p>例えば、<b>【サーバーログの保持期間】</b> を 30 日にすると、プロセスサーバーは過去 30 日間のサーバーログデータを保持し、それより以前のサーバーログデータは削除します。</p>

メンテナンススケジュールの設定後、**【更新】** **【保存】** を選択して設定を保存します。

## システムサービス

### 警告サービス

フォールトが原因でプロセスが停止したときに実行する電子メールまたはカスタムサービスを設定できます。

警告サービスは、Secure Agent またはクラウドサーバー上で実行するプロセスに対して使用できます。

警告を受信するには、**【サーバーのプロパティ】** で説明されている **【フォールトをキャッチしなかったときにプロセスを一時停止する】** プロパティを設定するか、Process Designer で定義したプロセスに対してこのオプションを有効にします。

警告サービスは、アプリケーション統合コンソールで選択するクラウドサーバーまたは Secure Agent に固有のものです。Informatica Cloud を使用する際、クラウドサーバー用の警告サービスを 1 つ、Secure Agent ごとに警告サービスを 1 つ設定して警告を受信できます。

警告サービスの設定時に 2 つのオプションのいずれかを選択できます。

- 電子メールの送信** このオプションを使用すると、一時停止またはフォールトプロセスの状態が発生したときに、指定した 1 人の受信者に電子メールを送信できます。このオプションを使用するには、電子メールサービスを設定する必要があります。

- **カスタムサービス** このオプションを使用すると、カスタムプロセスをデプロイできます。例えば、プロセスフォールトの管理者に通知を行うプロセスをデプロイできます。カスタムサービスは、次の方法で作成できます。

- Process Developer を使用する場合は、カスタムサービスとともに使用する PDD ファイルに表示されるサービス名を【サービス】フィールドに指定します。このサービス名は、アプリケーション統合コンソールの【サービス定義】ページに表示されます。カスタム警告サービスの定義方法の詳細については、『*Process Developer*』のヘルプを参照してください。

- Process Designer を使用する場合は、プロセスを作成し、次のステップを実行します。

1. 【イベントの開始】を【イベント】に設定します。
2. 【イベントソース名】の横の【システムイベント】を選択してから、【実行時警告サービス】を選択します。

**注:** Process Designer では、実行時警告サービスの開始イベントがある複数のプロセスを定義できます。ただし、警告サービスを設定したプロセスを 1 つ定義します。

警告サービスを使用するには【警告サービスの有効化】を選択し、【更新】をクリックして変更を保存します。

## 電子メールサービス

キャッチされないフォールトが原因でプロセスが一時停止したときにトリガする警告サービスなど、電子メールサービスを設定することができます。電子メールサービスの設定により、指定した管理者に電子メールを送信できます。

Process Designer で設計したプロセスを実行する場合は、アプリケーション統合コンソールを使用して、Secure Agent で実行するプロセスに使用する電子メールサービスと、Cloud Server で実行する組織のプロセスに使用する電子メールサービスを設定できます。次に、キャッチされないフォールトが原因でプロセスが一時停止されたときに通知を送信する警告サービスを定義するときに、【電子メールの送信】オプションを使用します。

電子メールサービスを定義するには、次のプロパティを設定します。

プロパティ	説明
SMTP サービスの有効化	すべてのプロパティを設定した後に、電子メールサービスを有効にする場合に選択します。
ホスト	電子メールサーバーの DNS 名 (mail.mydomain.com など) または IP アドレス (192.168.1.1 など) を入力します。
ポート	プロセスサーバーと電子メールサーバーとの間の通信に使用するポートを入力します。デフォルト値は 25 です。
認証タイプ	パスワードベースの認証または OAuth 認証を選択します。

パスワードベースの認証を使用して電子メール接続を作成するには、次のプロパティを設定します。

プロパティ	説明
ユーザー名	必須。電子メールサーバーへのログインに使用する名前（通常はアカウント名）を入力します。このフィールドは、SMTP サーバーで資格情報が不要な場合はオプションとなります。 例: notifyme@mydomain.com
パスワード	必須。ユーザー名のパスワードを入力して確定します。
セキュリティ	(オプション)。必要に応じて、セキュリティプロトコルを選択します。 - TLS (トランスポート層セキュリティ) - SSL (セキュアソケットレイヤ) 適切なセキュリティトランスポートを有効にするようにポートが設定されていることを確認してください。

OAuth 認証を使用して電子メール接続を作成するには、接続作成ページで次のプロパティを設定します。

プロパティ	説明
認証 URL	必須。ユーザー要求の認証に使用する電子メールサービスの OAuth 認証 URL を入力します。 例: <a href="https://login.microsoftonline.com/xxxxxx-xxxx-xxxx-xxxxxxxxx/oauth2/v2.0/authorize">https://login.microsoftonline.com/xxxxxx-xxxx-xxxx-xxxxxxxxx/oauth2/v2.0/authorize</a>
トークン要求 URL	必須。トークン要求を処理する OAuth トークン要求 URL を入力します。 例: <a href="https://login.microsoftonline.com/xxxxxxxx-xxxx-xxxx-xxxxxxxxx/oauth2/v2.0/token">https://login.microsoftonline.com/xxxxxxxx-xxxx-xxxx-xxxxxxxxx/oauth2/v2.0/token</a> リフレッシュトークンの有効期限は 90 日です。ユーザーは、トークンの有効期限が切れる前に、再度認証して接続をパブリッシュする必要があります。
クライアント ID	必須。OAuth プロバイダからの識別子の値を指定します。
クライアントシークレット	必須。電子メールアプリケーションに接続するためのクライアントシークレットを入力します。
スコープ	必須。スコープを指定します。OAuth 認証のスコープは、アプリケーションからユーザーのアカウントへのアクセスを制限するものです。1 つのクライアントに対して複数のスコープを選択できます。複数のスコープを入力するには、各値をスペースで区切ります。 Microsoft Outlook 電子メールアカウントの場合は、次のスコープを入力します。 <a href="https://outlook.office.com/SMTP.Send offline_access">https://outlook.office.com/SMTP.Send offline_access</a>
承認ステータス	現在のステータス、承認したユーザーの名前、最後に承認が完了した時刻を示します。
アクセスの承認	OAuth を使用した承認ワークフローを開始するにはクリックします。

設定をテストし、保存します。

## シェルサービス

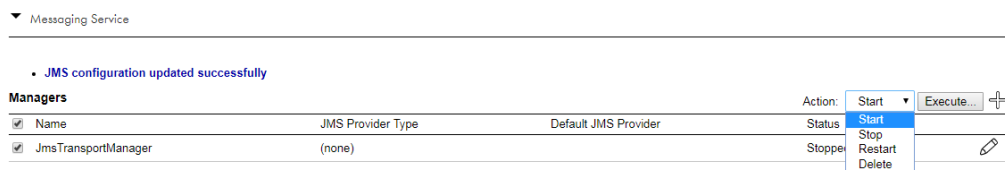
[シェルサービス] オプションを使用すると、Secure Agent で実行するプロセスについて、サービス呼び出しステップにシェルコマンドを含むプロセスを定義できます。

シェルサービスを有効にしたら、サービスステップを含むプロセスを作成し、システムサービスとしてシェルコマンドを追加します。

## メッセージサービス

このタブは、Secure Agent を選択した場合にのみ表示されます。

このページを使用して 1 つ以上の JMS Messaging Service のパラメータを設定すると、プロセスサーバーは JMS プロバイダへの接続を確立できます。



メッセージサービスの実行を制御するには、[アクション] リストを使用します。最初にリストでマネージャ名を選択してから、アクション（起動、停止、再起動、削除）を選択します。次に **【実行】** をクリックします。

## メッセージサービスの設定

メッセージサービスを設定するには、**【追加】** をクリックし、次のプロパティを入力します。

### デフォルトの JMS プロバイダ

必要に応じ、[JMS プロバイダタイプ] で選択したプロバイダについて、このプロバイダをデフォルトとして有効にします。複数のサービスを設定している場合は、1 つのプロバイダをデフォルトとして選択できます。

### JMS プロバイダタイプ

使用中の JMS プロバイダタイプの事前定義済みデフォルト設定を含む、設定テンプレートのリスト。理論上、JMS リソースへの JNDI アクセスを提供するプロバイダはどれでも使用できます。特定のプロバイダがリストにない場合は、**【他の JMS】** を選択し、よく使用される汎用的な JNDI プロパティを使用して設定を取り込みます。

### 接続ファクトリ名

JMS 接続ファクトリの JNDI 名。

### 接続ユーザー

接続ファクトリで接続を作成する際にユーザー資格情報を含めます。アプリケーションサーバーがホストするプロバイダを使用する場合、通常これは必要ありません。認証は JNDI コンテキストを介してオブジェクトにアクセスする際に行われるためです。MQ Series は、接続の資格情報を必要とするプロバイダの 1 つです。

### 接続パスワード

上記のユーザーのパスワード。

### 空の資格情報の送信

MQ Series の場合、認証が有効でないときにリモートキューマネージャに接続することがあります。その場合は、接続で空の文字列 ("") をユーザー名とパスワードの代わりに送信する必要があります。そうしないと、セキュリティ例外で失敗します。

## 最大合計接続数

各 JMS マネージャは、パフォーマンス向上のため、リモート JMS プロバイダとの相互運用時に内部的な接続プールを維持します。

一度に許可するアクティブな接続の最大数を設定します。これには、非同期リスナに使用される接続（[キューおよびリスナ]と[トピックおよびリスナ]で設定）、および呼び出しアクティビティに使用される接続が含まれます。最大数に達した場合、クライアントは接続がプールに返されるまで待機する必要があります。

最大値を-1 に設定すると、マネージャは必要な数だけ無制限に接続を作成できるため、クライアントは待機する必要がありません。JMS プロバイダが処理できる数を超える接続が作成された場合は、この値をデフォルトの-1 から変更します。

この値は、次のいずれかに設定する必要があります。

- -1（無制限）
- キューとトピックのリスナに、呼び出し用の接続を足した合計数より多い数。

## 空き接続の最大数

マネージャが一度に保持する未使用接続の数を設定します。接続がプールに返され、すでに最大数の空き接続がアイドル状態の場合、接続は閉じられます。これにより、プールからの合計接続数を必要に応じて増減できます。空き接続の最大数を 0 に設定すると、プロセスサーバーが接続を保持することがなくなります。各クライアントは、新たに作成された接続を受信します。

空き接続の値は、[最大合計接続数] の値より低くする必要があります。

## 配信モード

この設定は、JMS プロバイダが、すべてのプロセスについて、メッセージをストレージに永続的に保持するかどうかを制御します。各プロセスには、この設定より優先される、別の永続性設定を個別に指定できます。

JMS エラーが発生してもメッセージを永続的に保持するには、この設定を有効にします。このモードが有効な場合、プロセスサーバーは、JMS プロバイダでエラーが発生しても送信中のメッセージが失われないことを保証するように JMS プロバイダに指示します。これは、安定したストレージに記録されます。

永続的な配信では、JMS プロバイダにストレージが設定されている必要があります。また、メッセージを永続的に保持すると、通常は負荷が増します。

デフォルトでは無効です。これは、メッセージが非永続的な配信モードで送信されることを意味します。このモードでは、プロバイダにストレージが設定されているかどうかを把握する必要はありません。

## 有効期間（ミリ秒）

コンシュームされないメッセージをキューに残す時間を指定します。メッセージが一定期間後に無効になる場合は、有効期間を設定することをお勧めします。古くなったメッセージの有効期限が切れても、ストレージとコンピューティングリソースは維持されます。

デフォルトは 0、つまりプロバイダのデフォルト値が使用されます。通常、これはメッセージが有効期限切れにはならず、永続的にキューに残ることを意味します。

有効期間を設定する場合は、REST エンドポイントを使用してプロセスを呼び出します。HTTP メッセージが有効期間内にコンシュームされない場合、要求は失敗します。次のエラーメッセージで HTTP エラー（503 サービスが使用不可能になりました）が発生します。

定義された TTL を超えたメッセージが破棄されました。

## 優先度（int）

メッセージ処理の優先度として、負以外の整数を指定します。デフォルトは 0、つまりプロバイダのデフォルト値が使用されます。各プロセスには、この設定より優先される、別の優先度を個別に指定できます。

JMS は、0（最低）から 9（最高）までの 10 段階の優先度の値を定義します。クライアントは、0 から 4 までを標準的な優先度、5 から 9 までを上位の優先度と見なします。デフォルトでは、優先度は 4 に設定されています。

### 初期コンテキストプロパティ

JNDI ルックアップを介したアクセスが目的で、JMS リソースをホストするサーバーへの接続を確立するための名前/値ペアのセット。

例えば、プロバイダタイプとして「BEA Weblogic」を選択すると、WebLogic クライアントで一般的に使用されるデフォルトの初期コンテキストプロパティのセットが表示されます。URL、ユーザー名、パスワードの値は、使用する環境に合わせて更新します。

### キューおよびリスナ

JMS を介して外部のクライアントやサービスとの接続を有効にするには、外部 JMS サーバーとの接続を設定します。

- **キュー名/トピック名（必須）。**設定を説明する名前
- **JNDI の場所（必須）。**JNDI ルックアップで使用される場所の名前
- **リスナクラス（必須）。**メッセージリスナクラスの名前。リスナクラスは、プロセスサーバーへのメッセージのディスパッチを担当します。  
*Process Developer*で作成したプロセスをデプロイする場合は、次のリスナクラス名を使用します。  
`com.activeee.rt.mom.jms.transport.AeJmsBpelListener`  
*Process Designer*で作成したプロセスをデプロイする場合は、次のリスナクラス名を使用します。  
`com.activevos.rt.socrates.web.mom.jms.AeRealTimeJmsListener`  
必要に応じて、デフォルトのリスナクラスを拡張してカスタムの動作を組み込んだ、カスタムリスナクラスを作成できます。カスタムリスナはここで指定します。
- **リスナ数（必須）。**オープン状態を維持する接続の数。サーバーの起動時、JMS マネージャは、ターゲットで非同期コンシューマとして動作する、このクラスのインスタンスを作成します。各非同期コンシューマには、JMS サーバーに対する独自の接続があります。コンシューマの数および作成された接続の数は、リスナ数で制御されます。
- **セレクトラ（オプション）。**JMS メッセージセレクトラ文字列。
- **デフォルトサービス（オプション）。**アドレス指定ヘッダーまたはメッセージプロパティからターゲットサービスを判別できないときに使用する BPEL サービスの名前を指定します。  
**注:** サービスを指定する代替方法として、JMS メッセージプロパティの `JmsTargetService = "myRoleServiceName"`を受信メッセージに含めることができます。別の代替方法として、`myRole` パートナリリンクの `wsa:To` ヘッダーに、クエリパラメータとしてサービス名を含めることができます（例: `<destination JNDI name>?<servicename>`）。これらのシナリオは、JMS を使用して XML メッセージを受信するためのものです。
- **Run-As ID（オプション）。**JMS 要求の ID を指定します。ID はロールベースまたはグループのメンバーシップです。指定しない場合、要求は非同期的に実行されます。このプロパティの使用は、`ejb-jar.xml` でメッセージ駆動型 Bean を指定することに似ています。このオプションは、プロセスデプロイメント記述子ファイル（`pdd`）の「許可されたロール」制限で保護されたプロセスを JMS リスナが呼び出すようにする場合に使用します。別の使用事例は、ヒューマンタスク（B4P）操作に `aeRunAs` ヘッダーを使用する場合です。この事例では、`RunAs ID` を `abTrust` セキュリティロールとともに使用できます。
- **テナントコンテキスト。**マルチテナントライセンスを持つサーバーに適用されます。
- **XA トランザクション（オプション）。**サポート対象プラットフォームで有効になっており、必要なセットアップが実行されている場合は、キューから読み取られる JMS とプロセスエンジンが同じトランザクションに参加することを示します。詳しくは、[「XA トランザクション処理の有効化」](#)（ページ 42）を参照してください。

- **エラー時にロールバック。**【エラー時にロールバック】は、例外を受け取ったメッセージまたは失敗したメッセージを却下し、次のメッセージに進む場合に選択します。例えば、認証の失敗により例外が返される場合があります。

【エラー時にロールバック】を選択する場合は、メッセージサーバーによる失敗したメッセージの処理方法を設定する必要があります。例えば、失敗したメッセージの再送信を 10 回行うようにメッセージサーバーを設定できます。または、失敗したメッセージをデッドレターキューに移動し、後で処理方法を決定するようにサーバーを設定することもできます。

例えば、次のポリシーエントリを追加して、Apache ActiveMQ サーバーが失敗したメッセージをデッドレターキューに移動するように設定できます。

```
<policyEntry queue="">
  <deadLetterStrategy>
    <individualDeadLetterStrategy queuePrefix="DLQ." useQueueForQueueMessages="true"
processNonPersistent="true"/>
  </deadLetterStrategy>
</policyEntry>
```

ここでは、元のキューが userIDs、Apache ActiveMQ サーバーが作成するデッドレターキューが DLQ.userIDs を表します。

【エラー時にロールバック】オプションは、ターゲットプロセスに配信されたメッセージには適用されません。メッセージは、ターゲットプロセスを呼び出すと、プロセスが失敗しても正常に処理されたと見なされます。

新規のキューおよびリスナでは、【エラー時にロールバック】オプションはデフォルトで選択されています。

### トピックおよびリスナ

このセクションでは、プロセスサーバーの代わりにメッセージを受信するリスナを設定します。受信メッセージをサーバーにディスパッチするため、サーバー上の JNDI の場所にバインドされたリスナが最低でも 1 つ必要です。

より多くの宛先がサービス要求に必要な場合は、【キューの追加】または【トピックの追加】を選択して、新しい定義を追加できます。

## Websphere ユーザー

IBM WebSphere をプロバイダとして選択していて、プロセスサーバーが WebSphere で実行中の場合、キューテーブルとトピック構成テーブルは表示されません。

WebSphere JMS プロバイダではキューリスナとトピックリスナの構成は使用できません。WebSphere JMS では、WebSphere サーバーを実行中に Message-Driven Bean のデプロイメント外部で非同期コンシューマを作成することが明示的に禁じられています。

プロセスサーバーは、Message-Driven Bean デプロイメント向けのサンプルアプリケーションを提供しています。

## XA トランザクション処理の有効化

インバウンド要求に JMS トランSPORTを使用している場合、1 回のみのメッセージ配信を確実に保証するための唯一の方法は、プロセス状態をプロセスサーバーデータベースにコミットし、キューから JMS メッセージを読み取る分散 (XA) トランザクションを使用することです。XA トランザクションを使用しないと、極めて稀であるものの、インバウンド受信が Process Server データベースにコミットされたときにサーバーがダウンすると、重複する要求は受け取るが JMS トランザクションはコミットされていないという事態が発生する場合があります。この場合、メッセージはキューに残ったままになり、再試行される場合もありますが、重複するプロセス要求が発生します。

JMS Messaging Service の分散 (XA) トランザクション処理は、プロセスで必要な場合は有効にできます。この機能を有効にするには、以下の「必要な設定」に示す、いくつかの手動による手順が必要です。

**注:** XA トランザクションは比較的負荷が高くなる場合があるため、有効にするとパフォーマンスが低下する可能性があります。

分散トランザクションを使用すると、プロセスサーバーはメッセージキューからメッセージを取得し、ACID 特性（原始性、一貫性、独立性、永続性）を持つ単一のトランザクション単位でデータベースを更新します。この機能により、JMS から扱われるメッセージが 1 回のみ処理されることが保証されます。

### サポートされるプラットフォーム

XA トランザクション処理は、JTA トランザクションマネージャを含む、すべてのプロセスサーバーアプリケーションサーバーでサポートされます。唯一の例外は Apache Tomcat です。

すべてのプロセスサーバーデータベース（MySQL、DB2、Oracle、および SQLServer）はサポートされます。

**注:** 一部のデータベースでは、XA サポートがデフォルトで有効になっていません。このサポートは、データベース管理者が有効にする必要があります。XA トランザクションが有効かどうかを確認し、どのように有効にするかについては、データベースのドキュメントを参照してください。

### 必要な設定

XA トランザクションを設定するには、以下のガイドラインに従います。

1. データベースが XA トランザクションをサポートするように設定されていることを確認します。
2. XA トランザクションをサポートする別の JDBC データソースを作成します。データソース設定の詳細については、データベースおよびアプリケーションサーバーのドキュメントを参照してください。デフォルトでは、プロセスサーバーは jdbc/ActiveVOSXA という JNDI 名のデータソースを検索します。  
**注:** プロセスサーバーは、XA トランザクション用の別のデータソース設定を使用します。XA トランザクションと非 XA トランザクションを同じ接続プールに混在させると問題が発生する可能性があるためです。
3. アプリケーションサーバーがホストする JTA トランザクションマネージャを登録する、XA 対応の JMS 接続ファクトリを設定します。詳細については、JMS プロバイダのドキュメントを参照してください。アプリケーション統合コンソールにある JMS Messaging Service のページでこの接続ファクトリを指定します。
4. 新しいメッセージサービスを作成するときに表示される [メッセージサービス] ウィンドウの [XA トランザクション] ボックスを選択します。

## データソースサービス

このサービスは、Secure Agent を選択した場合にのみ表示されます。

多くのプロセスは、リレーショナルデータベースと対話します。データソースサービスを使用すると、アプリケーションの現在の全ユーザー間でオープンになっている接続の「プール」を作成できます。ある時点で実際に要求を実行しているユーザーの数は通常、アクティブユーザーの総数に比べて非常にわずかです。また、要求が処理されているときのみ、データベース接続が必要となります。ここでは、DBMS にログを記録し、ユーザーアカウントの問題を処理するサービスを作成します。

**注:** データソースサービスは、Apache Commons DBCP に基づいています。詳細については、<http://commons.apache.org/proper/commons-dbcp/index.html> を参照してください。

新しいデータソースサービスを追加するには、**[新規]** をクリックします。

データソースサービスを設定するには、次のプロパティを入力します。

## Datasource Service

Standard Properties	
Name*	<input type="text"/>
JNDI Path*	<input type="text"/>
Driver Class*	<input type="text"/>
URL*	<input type="text"/>
Username	<input type="text"/>
Password	<input type="password"/>
Advanced Properties	
Initial Size	<input type="text"/>
Max Active	<input type="text"/>
Max Idle	<input type="text"/>
Max Wait (ms)	<input type="text"/>
Min Idle	<input type="text"/>
Default Transaction Isolation	<input type="text" value="(default)"/>
Pooling Statements	<input type="checkbox"/>
Min Evictable Idle Time (ms)	<input type="text"/>
Number Tests per Eviction Run	<input type="text"/>
Time Between Eviction Runs (ms)	<input type="text"/>
Test on Borrow	<input type="checkbox"/>
Test on Return	<input type="checkbox"/>
Test while Idle	<input type="checkbox"/>
Validation Query	<input type="text"/>
Custom Connection Properties	
Name	Value

変更を加えたら、[テスト接続] をクリックして、接続が正しく定義および設定されていることを確認します。

### 標準プロパティ

標準プロパティは次のとおりです。

#### 名前

定義または編集するサービスの名前。

#### ドライバクラス

使用する JDBC ドライバの完全修飾された Java クラス名。

#### JNDI パス

このデータソースをバインドする JNDI 名。

#### パスワード

接続を確立する際に JDBC ドライバに渡す接続パスワード。

URL

接続を確立する際に JDBC ドライバに渡す接続 URL。

### ユーザー名

接続を確立する際に JDBC ドライバに渡す接続ユーザー名。

### 詳細プロパティ

詳細プロパティは次のとおりです。

#### デフォルトのトランザクション分離

このプールによって作成される接続のデフォルト状態。次のいずれかになります。

- なし
- READ\_COMMITTED。
- READ\_UNCOMMITTED
- REPEATABLE\_READ
- SERIALIZABLE

#### 初期サイズ

プールが開始されたときに作成される初期接続の数。

#### 最大アクティブ数

このプールから同時に割り当てることが可能なアクティブな接続の最大数。負の値の場合、制限はありません。

#### 最大アイドル数

別の接続を解放せずにプール内でアイドル状態を維持できる最大接続数。負の値の場合、制限はありません。

#### 最大待機時間（ミリ秒）

使用可能な接続がない場合に、プールが接続を待機する最大ミリ秒数。値-1 は、無限に待機することを意味します。

#### エビクト可能な最小アイドル時間（ミリ秒）

オブジェクトがプール内でアイドル状態でいられる最小時間。この時間を過ぎると、アイドルオブジェクトエビクタによるエビクト（退去）の対象になります（エビクタが存在する場合）。

#### 最小アイドル時間

別の接続を作成せずにプール内でアイドル状態を維持できる最小接続数。値ゼロは何も作成されないことを意味します。

#### エビクト実行ごとのテスト数

アイドルオブジェクトエビクタスレッド（ある場合）の実行時に調べるオブジェクト数。

#### プーリングステートメント

このプールに対し、用意されているステートメントプーリングを有効にします。

#### 貸し出し時にテスト

これを選択すると、オブジェクトはプールから貸し出される前に検証されます。オブジェクトを検証できない場合はプールから破棄され、別の貸し出しが新たに試みられます。

#### 返却時にテスト

これを選択すると、オブジェクトはプールに返却される前に検証されます。

### アイドル状態時にテスト

これを選択すると、オブジェクトはアイドルオブジェクトエビクタ（ある場合）によって検証されます。オブジェクトを検証できない場合はプールから破棄されます。

### エビクタ実行間隔（ミリ秒）

アイドルオブジェクトエビクタスレッドの次回実行時までスリープする時間（ミリ秒）。これが負の値の場合、アイドルオブジェクトエビクタスレッドは実行されていません。

### 検証クエリ

呼び出し元に返す前にこのプールの接続を検証する SQL クエリ。指定する場合、クエリは 1 つ以上の行を返す SQL SELECT ステートメントである必要があります。

### カスタム接続プロパティ

このセクションを使用すると、接続に必要な他のプロパティを追加できます。この情報は、サービスが接続を作成するときに JDBC ドライバに送信されます。**[追加]** ボタンをクリックして、プロパティの名前と値を入力する行を追加します。

## 第 3 章

# デプロイされたアセット

【デプロイされたアセット】 ページでは、コントリビューション、デプロイされたプロセス、カタログリソース、インデックス付きプロパティ、URN マッピングを表示および管理できます。

## ビジネスプロセスアーカイブのデプロイ

【デプロイ】 ダイアログでは、アプリケーション統合コンソールを使用して、ビジネスプロセスアーカイブ（.bpr ファイル）を Cloud Server または Secure Agent に追加できます。Process Developer で作成される .bpr ファイルは Web アーカイブと似ており、実行時に利用できるようにプロジェクトリソースのデプロイに必要なリソースをすべて含んでいます。

.bpr アーカイブは 1 つずつデプロイしますが、.bpr にすべての BPEL ファイル、デプロイメント記述子、パートナー定義ファイル、WSDL 定義、その他の必要なリソースを含めることができます。

.bpr ファイルをデプロイするには、次の手順を実行します。

1. 管理権限のあるアカウントを使用して、Informatica Intelligent Cloud Services<sup>SM</sup>にログインします。
2. アプリケーション統合コンソールを選択します。
3. 【デプロイされたアセット】 を選択します。
4. ページの右上隅にある【デプロイ】 をクリックします。
5. 【参照】 をクリックして、対象の .bpr ファイルを選択します。
6. クラウドサーバーまたは Secure Agent を【ターゲットのデプロイ】 リストから選択します。
7. 【デプロイ】 をクリックします。

サーバーが、.bpr に含まれるファイルを検証し、データベース内にファイルを格納します。【ログ】 ページに、デプロイされたプロセスファイルに関するエラー、警告、情報が表示されます。

.bpr ファイルをデプロイ後、デプロイメント記述子、パートナー定義ファイル、BPEL ファイル、インデックス付きプロパティ、WSDL 定義、およびスキーマファイルの詳細を【デプロイされたアセット】 ページ【カタログ】 の特定のリソースにアクセスすることで表示できます。を選択します。

## コントリビューションについて

【コントリビューション】 タブには、すべてのビジネスプロセスアーカイブ（BPR）のデプロイメントがリストされます。それらの名前は、デプロイ元の Process Developer プロジェクトに基づいています。

Deployed Assets
 

bs3e1cdb1z-Agent ▼

Deploy

Contributions	Deployed Processes	Resources	Indexed Properties	URN Mappings
Contributions [123]		Updated 8:37 AM UTC		
Name	Contribution Path	Version	State	Publish Date
<a href="#">DAS_Agent</a>	project:	1.0	ONLINE	2018/10/16 06:15 PM
<a href="#">DAS_Common</a>	project:	1.0	ONLINE	2018/10/16 06:15 PM
<a href="#">NotifyAssessModificationStub</a>	project:	1.0	ONLINE	2018/08/07 03:01 AM

Process Developer がすべてのファイルを 1 つの Orchestration プロジェクトにまとめるのと同じように、コントリビューションはプロジェクトのデプロイ可能なプロセスとリソースをまとめた 1 つの単位です。これは、サーバーカタログへの一意のデプロイメントです。開発者は、プロジェクトを更新するときに、以前のコントリビューションをサーバーに維持したまま、新しいコントリビューションをデプロイできます。

コントリビューションの特長を以下に示します。

- 各コントリビューションには、Process Developer の場所のヒントとして、デフォルトで project:/project name という名前が付けられます。
- コントリビューションには複数のプロセスを含めることができ、それらは個別に管理することもまとめて管理することも可能です。
- 新しいバージョンのコントリビューションは前のバージョンのコントリビューションより優先され、現在の（オンラインの）バージョンになります。ただし、オンラインのコントリビューションを以前のバージョンにロールバックすることは可能です。このコントリビューション追跡システムにより、プロセスとそのリソースを簡単に管理できます。
- 現在の（オンラインの）コントリビューションは 1 つのみです。すべてのコントリビューションはオンラインバージョンになれます。
- 現在の（オンラインの）コントリビューションは、前のバージョンにロールバックできます。
- プロセスのバージョンは、WSDL、スキーマ、その他のリソースの特定のバージョンに関連付けられています。すべての BPEL プロセスは、デプロイされた一連のリソースに関連付けられています。
- コントリビューションを削除すると、関連付けられたファイルとデプロイメントログもすべて削除されるため、この方法でテスト中にデータベースを簡単にクリーンアップできます。
- 複数の開発者が、競合なく同じプロセスとリソースをデプロイすることもできます。

詳細については、「コントリビューションの詳細」を参照してください。

**コントリビューションが更新された場合**

BPR コントリビューションが最初にデプロイされる際、プロセスサーバーはコントリビューションとデプロイメント内のすべてのファイルをバージョン 1.0 としてにタグ指定し、コントリビューションはオンラインになります。その後開発者が、ファイルを更新、追加、または削除して変更を加え、同じ BPR を再デプロイします。これで新しいバージョンがオンラインバージョンになります。古いバージョンはオフライン保留となり、実行中のすべてのプロセスが完了するとオフラインになります。

次のシナリオは、コントリビューションが更新された場合に何が起きるかを示しています。

V1 から V2 へのコントリビューションの変化 -- Process1 を変更、Process2 を削除

デプロイメント 1 project:/MyProject V1	デプロイメント 2 project:/MyProject V2	サーバー上の変更
process1.bpel	process1.bpel (変更あり)	V2 はオンライン、V1 はオフラインまたはオフライン保留です (関連付けられたプロセスインスタンスがある場合)。
process.private.wsdl process.public.wsdl	変更なし	V1.01 はオンラインです。V1 はオフラインです。各デプロイメントで、WSDL と XSD はキャッシュの問題を回避するために増分されます。
process2.bpel	(含まれない)	プロセスはカタログに存在しなくなりました。

V1 から V2 へのコントリビューションの変化 -- 元のファイルをすべて削除、新規をデプロイ

デプロイメント 1 project:/MyProject	デプロイメント 2 project:/MyProject V2	サーバー上の変更
(myAVOS.avcconfig は存在しない)	myAVOS.avcconfig	ファイルは「追加リソース」としてデプロイされます。これは、前のデプロイメントと同じ BPR にデプロイされる唯一のファイルです。V1 はオンラインです。これは、コントリビューション V2 で唯一のファイルなので、オンライン状態の唯一のファイルです。すべてのファイルをオンラインに戻すには、BPR を再デプロイしてすべてのファイルを含めます。
process1.bpel	(未デプロイ)	BPEL ファイルは削除されます。
process.private.wsdl process.public.wsdl	(未デプロイ)	WSDL ファイルは削除されます。

各コントリビューションについて、次の表に示すように、基本的なバージョン情報が表示されます。


項目	説明
コントリビューション	ベース URI。通常は project:/myProject などの Process Developer プロジェクト名。
バージョン	デプロイされたバージョン
状態	デフォルトでは、オンライン状態が表示されます。詳細については、この表の下にある全状態の説明を参照してください。
日付	コントリビューションがデプロイされた日付。
グループ	Process Developer のビジネスプロセスのエクスポートウィザードで追加する、このコントリビューションが属するグループ (ある場合)。
デプロイヤー	デプロイを実行した人物またはシステムの名前 (デプロイヤーに認証が必要な場合)。デフォルトは匿名です。

## コントリビューションのバージョン

- オンラインデフォルトでは、最新の（番号が一番大きい）コントリビューションがオンライン（現在）のバージョンです。
- オフライン保留バージョンをオフラインに設定した状態で実行中のプロセスインスタンスがある場合、実行中のプロセスインスタンスがあるときに新しいバージョンがデプロイされた場合、またはコントリビューションのエクスポートされた名前空間が他のコントリビューションから参照されている場合、バージョンは「オフライン保留」状態になります。
- オフライン—オフライン保留バージョンのプロセスインスタンスがすべて完了すると、コントリビューションはオフラインになります。

次の選択フィルタを選択すると、コントリビューションのサブセットを表示できます。

- ## コントリビューションの詳細


bpr-airline

Id: 133863217213476864

Tenant: 0CBIPXmws9sdkKh9w443Sk

Contribution: 0CBIPXmws9sdkKh9w443Sk/bpr-airline

Version: 1.0

State: ONLINE

Group:

Description:

Set to Offline

Delete...

Deployed Processes

Name	Target Namespace	Version	State	Group
<a href="#">airlineCompleteDynamic</a>	http://airlineCompleteDynamic	1.0	ONLINE	
<a href="#">cancel</a>	http://cancel	1.0	ONLINE	
<a href="#">creditLookUp</a>	http://creditLookUp	1.0	ONLINE	
<a href="#">flightLookUp</a>	http://flightLookUp	1.0	ONLINE	
<a href="#">return_address</a>	http://return_address	1.0	ONLINE	
<a href="#">setReservation</a>	http://setReservation	1.0	ONLINE	

Contributed Resources

Name	Version	Online	Target Namespace	Group
<a href="#">address.wsdl</a>	1.0	Yes	http://return-address.activebpel.org	
<a href="#">address.xml</a>	1.0	Yes		

コントリビューションの状態がオンラインの場合、コントリビューションは現在のバージョンであり、**【オフラインに設定】**を選択できます。他のコントリビューションバージョンがある場合は、1つ上または下のバージョンがオンラインに設定されます。コントリビューションの BPEL ファイルに対してプロセスインスタンスが実行中の場合、コントリビューションは**【オフライン保留】**に設定されます。すべてのプロセスインスタンスが完了すると、状態はオフラインに変わります。

コントリビューションの状態がオフラインの場合は、**[オンラインに設定]**を選択できます。コントリビューションのファイルは現在のバージョンになります。オンライン日（有効日）に BPEL がデプロイされると、プロセスの状態はオンライン保留になります。

コントリビューション内の BPEL プロセスの状態はコントリビューション自体の状態と同じですが、オンラインのコントリビューションから BPEL プロセスを個別に選択して状態を変更することができます。例えば、オンラインコントリビューションに 3 つのプロセスがある場合、1 つ、2 つ、または 3 つすべてのプロセスをオフラインにできます。

### コントリビューションの削除

カタログからファイルのユニット全体を削除するには、**[削除]**を選択します。実行中のプロセスが存在する場合、コントリビューションはプロセスが完了するまでオフライン保留状態になります。

コントリビューションを削除すると、次のイベントが発生します。

- コントリビューション内のすべてのファイルが削除されます。
- デプロイメントログが削除されます。
- コントリビューションにアタッチされた、完了済みのすべてのアクティブなプロセスが削除されます。

### コントリビューションの詳細情報

このページの情報は、以下に説明する複数のセクションにグループ化されます。

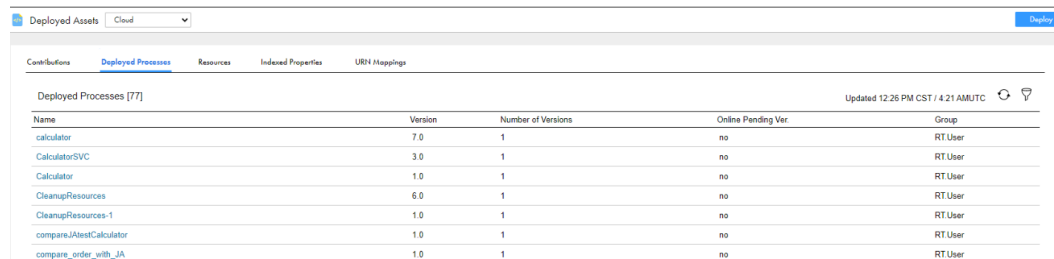
- **コントリビューションの詳細**  
各コントリビューションにはデプロイメント ID が割り当てられます。名前、状態、バージョン、およびグループの詳細は、**[コントリビューション]** のリストページに示す内容と同じです。**[説明]** は、指定されている場合、Process Developer の BPR エクスポート中に使用できます。
- **デプロイされたプロセス**  
BPEL ファイルのリスト（ある場合）と、それらの現在の状態。
- **コントリビューション対象リソース**  
コントリビューションにデプロイされるファイルのリスト。これらのファイルの例は、WSDL、スキーマ、HTML、XSL、XML、CSS、画像、i18N プロパティファイル、Process Central ファイル、イベントドキュメント、Java jar、およびレポート定義です。バージョンは、このコントリビューションと関連付けられた更新バージョンです。  
  
WSDL とスキーマは、前回のバージョンと同一でも、デプロイ時に必ず更新されます。他のリソースでバージョン番号が増えるのは、そのコンテンツが変更された場合のみです。  
  
これらのリンクのいずれかをクリックすると、プロセスサーバーによってソースが表示されます（可能な場合）。例えば、XML または HTML が表示されます。
- **エクスポートされた名前空間**  
コントリビューション内のリソースによって公開される、他のコントリビューションが使用するための名前空間。
- **インポートされた名前空間**  
このコントリビューションが依存する他のコントリビューションの名前空間（ある場合）。このコントリビューションが他のコントリビューションに依存する場合は、最初にこのコントリビューションを削除します。
- **デプロイメントログ**  
このコントリビューションのデプロイメントログへのリンク。

# デプロイされたプロセス

【デプロイされたプロセス】タブには、クラウドまたは Secure Agent にデプロイされたプロセスのリストが表示されます。

注: 各プロセスインスタンスのエントリが表示される【プロセス】ページとは異なり、【デプロイされたプロセス】タブには、作成された各プロセスのエントリが表示されます。

次の図に、【デプロイされたプロセス】タブの例を示します。



Name	Version	Number of Versions	Online Pending Ver	Group
calculator	7.0	1	no	RT User
CalculatorSVC	3.0	1	no	RT User
Calculator	1.0	1	no	RT User
CleanupResources	6.0	1	no	RT User
CleanupResources-1	1.0	1	no	RT User
compareJAtestCalculator	1.0	1	no	RT User
compare_order_with_JA	1.0	1	no	RT User

デプロイされたプロセスの名前を選択すると、[デプロイ済みプロセスバージョン詳細] ページが表示されます。詳細については、[Process Version Detail \(ページ 89\)](#)を参照してください。

## リソース

【リソース】タブには次のリソースタイプがあります。


リソースタイプ	説明
一元的な設定（オンプレミスのみ）	Process Central のタスク、フォーム、レポートについて、フォルダの表示や他のフィルタを管理する設定ファイル。
関数コンテキスト	プロセスにインポートされる POJO カスタム関数。
HTML ドキュメント	プロセスを開始するためのフォーム、およびユーザーのアクティビティのためのタスクフォーム。Process Central で使用されます。
画像	HTML ページで参照される画像ソースファイル。
Java Jar（オンプレミスのみ）	パートナーサービス用インタフェース。
レポート定義（オンプレミスのみ）	システム定義およびユーザー定義のレポート。
スキーマドキュメント	プロセスにインポートされる XSD。
WSDL ドキュメント	プロセスにインポートされる WSDL。
XQuery モジュール	プロセスにインポートされる XQuery カスタム関数。

リソースタイプ	説明
XSL ドキュメント	カスタム関数や古いインボックスアプリケーションのレンダリングのための変換型スタイルシート。
その他	<ul style="list-style-type: none"> <li>- カスタム関数、サンプルデータ、その他数々の設定ファイル用の XML ファイル</li> <li>- 多言語サポート用の.properties ファイル</li> <li>- フォーム用の JavaScript ファイル</li> </ul>



## カタログリソースの表示

リソースは、デプロイされたものの BPEL プロセスからもアクセスでき、1 コピーのみが維持されます。

次の図は、[リソース] タブを示しています。

 Deployed Assets
 bs3e1cdb1z-Agent ▼
Deploy

Contributions
Deployed Processes
**Resources**
Indexed Properties
URN Mappings

Resources [500]
Updated 8:43 AM UTC



**Total Reads:** 272581  
**Disk Reads:** 367  
**Cache Efficiency:** 99.87%  
**Cache Size:** 297 (0.01 MB)

Resource	Type	Target Namespace	Version	Last Modified	Group
✓ <a href="#">AcceptedEventRelation_detail.xml</a>	Other		1.0.11	2019/04/20 05:30 PM	Ce4sf. User
✓ <a href="#">AcceptedEventRelation_picklist.xml</a>	Other		1.0.11	2019/04/20 05:30 PM	Ce4sf. User
✓ <a href="#">Access_Token_detail.xml</a>	Other		1.0	2019/01/24 05:07 AM	RT. User

Secure Agent の [リソース] タブを表示すると、ノードがどのようにカタログにアクセスするかに関する次のような詳細情報が表示されます（Cloud Server の [リソース] を表示した場合、詳細情報は表示されません）。

項目	説明
合計読み取り数	プロセスの実行中にリソース情報を取得するための読み取りの数（キャッシュ内またはキャッシュ外）。
ディスク読み取り	絶対数と合計読み取り数の割合で表される、キャッシュ外のリソースファイルに対する読み取り数。
キャッシュ効率	メモリから読み取られるリソースとストレージから読み取られるリソースの割合。例えば、100%はすべてのリソースがメモリキャッシュから読み取られることを示します。
キャッシュサイズ	保存されたキャッシュ内のリソースファイルの数。デフォルトは 100 です。キャッシュサイズは、[管理] > [サーバーの設定] > [サーバーのプロパティ] で設定出来ます。キャッシュサイズを変更すると、エンジンのパフォーマンスが向上する場合があります。

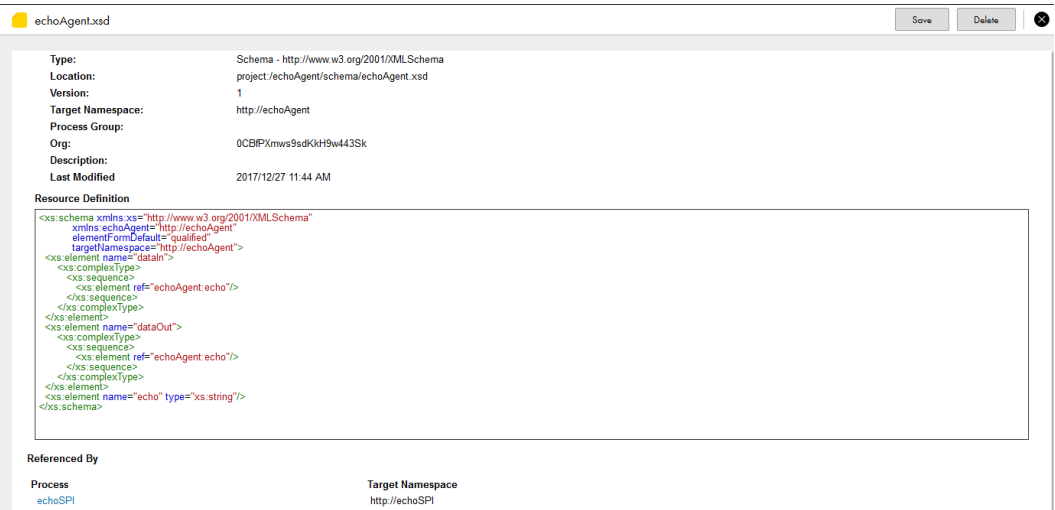
[リソース] リストには、リソースのタイプ、名前、および名前空間が表示されます。リソース名の上にマウスを重ねると、リソースのロード元の物理的な場所が表示されます。

このページに表示されるリソースの情報は、[リソース詳細] ページに表示される内容と同じです。リソースのリンクをクリックすると、[リソース詳細] ページが表示されます。

**[リソース詳細] ページ**

[リソース詳細] ページには、各リソースの [リソース] ページと同じ情報が表示されます。XML ソースコードも表示されます。

次の図は、[リソース詳細] ページの例を示しています。



プロパティ	説明
名前	リソースの名前
クラス	リソースをインスタンス化したクラス
場所	リソースのロード元となった物理的な場所。これは、デプロイメント記述子が作成されたときに場所を一意に定義するのに役立ちます。また、同じ名前の複数のリソースファイルをエンジンにデプロイするためにも使用出来ます。WSDL の場所は、.pdd ファイルで参照されます。
バージョン	リソースのバージョン。これは、プロセスサーバーによって自動的に作成されるか、.pdd ファイル内で割り当てられます。
ターゲット名前空間	リソースのターゲット名前空間
プロセスグループ	.pdd ファイルに追加されたグループ名。リソースを編成するために使用されます。
説明	.pdd ファイルに追加された説明
最終変更日	このリソースが最後に変更された日

[リソース詳細] ページには、このリソースを参照する、いくつかのリソースのプロセスバージョンが表示されます。

## 選択フィルタ


選択フィルタは次のように使用します。

- 状態（すべて、オンライン、またはオフライン）を選択します。詳細については、コントリビューションの説明を参照して下さい。
  - リストからリソースタイプを選択します。
  - リソース名を入力します。tutor\*.\*などのワイルドカードを使用出来ます。
  - ターゲットの名前空間を入力します。ワイルドカードを使用出来ます。
  - グループ名を入力します。デプロイメント中にすでにグループ名が定義されている場合は、検索出来ます。
- [送信] を選択して、フィルタ処理したリソース一覧を表示します。

# インデックス付きプロパティ

インデックス付きプロパティとは、アクティブなプロセスの選択フィルタとして機能する変数プロパティです。ビジネスイベント処理のイベントを定義する際に使用することもできます。このプロパティは、カスタム ID、申請日、金額などのデータを保持します。イベントサービスでインデックス付きプロパティを使用すると、ビジネスイベントがイベントマネージャで監視されたときに詳細な分析を行えます。選択クエリでインデックス付きプロパティを使用すると、重要なデータ項目に基づいてプロセスを素早くフィルタリングできます。

次の図は、[インデックス付きプロパティ] タブの例を示しています。

 Deployed Assets bs3e1cdb1z-Agent ▼ Deploy

Contributions	Deployed Processes	Resources	Indexed Properties	URN Mappings
Indexed Properties (7)				<input type="checkbox"/> Show Public <input type="checkbox"/> Show System
Plan Id	Name	Type	Variable Path	Query
1288	SOCRATES.correlationId	String	/process/variables/variable[@name='stack']	@correlationId
1288	SOCRATES.CurrentScreenTitle	String	/process/variables/variable[@name='LastKnownStepTitle']	
1288	SOCRATES.ExtraInfo	String	/process/variables/variable[@name='connectRequest']	types:context/n2:arg[@name='ExtraInfo']

例えば、同じインデックス付きプロパティを共有するフォールトしたプロセスのリストの取得、プロセスの一時停止、不良データ値の修正、プロセス実行の続行などを行うことができます。

インデックス付きプロパティは、プロセスデプロイメント記述子ファイルで定義されています。このテーブル内の情報は次のとおりです。

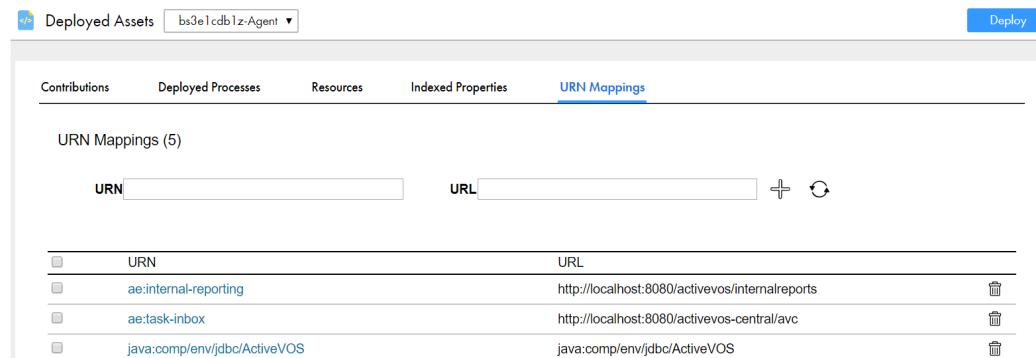
項目	説明
プラン ID	インデックス付きプロパティに関連するデプロイ済みプロセス。このリンクをクリックすると、「 <a href="#">[デプロイされたプロセスバージョンの詳細] ページ</a> 」 ( <a href="#">ページ 89</a> ) ページが表示されます。
名前	インデックス付きプロパティの名前。この名前は、選択フィルタ式ビルダのインデックス付きプロパティリストに表示されます。
タイプ	string や double などスキーマのタイプ。
変数パス	プロセスの変数名とプロセス内の宣言の場所。
部分	メッセージ型の変数のプロセス変数部分。
クエリ	プロセス変数部分の詳細（オプション）。

## URN マッピング

「URN マッピング」タブには、システムマッピングとユーザー定義のマッピングがあります。

Secure Agent を選択すると、4 つのデフォルトマッピングが表示されます。

次の図は、サンプルの「URN マッピング」タブを示しています。



### システムマッピング

プロセスサーバーは、システムサービスに対して次のデフォルトの URN マッピングを提供します。マッピングが初期設定に含まれていない場合は、手動による追加が必要になることがあります。

URN	URL
ae:internal-reporting	http://localhost:8080/activevos/internalreports レポートサービスを含むデプロイ済みプロセスのための、BIRT レポートエンジンのデフォルトアドレス。
ae:task-inbox	http://localhost:8080/activevos-central/avc Process Central のデフォルトアドレス。必要に応じ、インストール場所に合わせて host:port を変更します。
java:comp/env/jdbc/ActiveVOS	この値は、指定した JNDI データベース名を使用して作成されます。
urn:aeHostEnvironmentRuntime	avHostEnvironmentRuntimeAccess

### ユーザー定義のマッピング

URN (Universal Resource Name) には物理アドレスを割り当てることができます。URN はパートナーリンクの論理アドレスであり、デプロイメントリソースで指定されます。

URN マッピングは、ターゲットのエンドポイント参照を定義するための、柔軟で動的な方法を提供します。URN マッピングを使用すると、パートナーリンクのエンドポイント参照の物理アドレスを指定できます。プロセスデプロイメント記述子 (pdd) ファイルまたは WSDL ファイルで指定したアドレスを使用する必要はありません。URN を URL にマッピングすることにより、静的に定義したエンドポイントアドレスの呼び出しに依存する必要がなくなります。URN マッピングにより、例えば、テスト環境と本番環境に同じ BPR ファイルをデプロイするという柔軟な対応ができます。

また、URL を指定する場合、その URL を別の URL にマッピングすることで置換できます。

次の例は、あるタイプの URN から URL のマッピングを示しています。

urn:localhost = http://localhost:8080/active-bpel/services/\${urn.3}

このマッピングは、次のパートナーリンクのアドレス情報を使用してプロセスがデプロイされるときに使用できます。

```
<partnerLink name="assessor">
  <partnerRole endpointReference="static"
    invokeHandler="default:Address">
    <wsa:EndpointReference xmlns:assessor="http://
      tempuri.org/services/loanassessor">
      <wsa:Address>urn:localhost:AssessRisk</wsa:Address>
      <wsa:ServiceName PortName=
        "SOAPPort">assessor:LoanAssessor</wsa:ServiceName>
    </partnerRole>
  </partnerLink>
```

プロセスサーバーの呼び出しフレームワークは、次のように URN を解決します。

urn:localhost:AssessRisk = http://localhost:8080/active-bpel/services/AssessRisk

次に示すのは、URN から URL へのマッピング方法です。URN の各セグメントは、コロンで区切られています。

URN	URL
urnSegment1:urnSegment2	http://localhost:8080/active-bpel/services/MyService
http://ServerA:8080/active-bpel/services/MyService	http://ServerB:8081/active-bpel/services/MyService
urn:localhost:service	http://localhost:\${AE-NODE1-PORT}/active-bpel/services/\${urn.4}

上の表内の最後の例は、URL で変数代入を使用する方法を示しています。

URL の値には変数を含めることもできます。変数に使用できるのは、`java.lang.System.getProperties()`を介した環境変数または URN 自体からのセグメントです。Apache Ant スタイルの変数宣言である`${property}`は、URL 内のプロパティを特定するために使用されます。入力 URN 値からのセグメントは、特別なプロパティ命名規則である`${urn.offset}`を使用することで参照できます。`offset` は、代入に使用する入力 URN 値からのセグメントを識別するワンベースオフセットです。

この表の 3 つ目のマッピングにある URL には、2 つの変数が含まれています。`${AE-NODE1-PORT}`変数は、環境変数からポート番号を引き出します。この変数は、Java 実行環境で `-D` パラメータとして設定するか（例: `java -D AE-NODE1-PORT =8080 ...`）またはプロセスサーバーに対して外部的に指定する必要があります。

この例にあるもう 1 つの`${urn.4}`変数は、入力 URN 値から 4 つ目のセグメントを参照します。ただし、URN に含まれるセグメントは 3 つのみです。`.pdd` ファイルの URN には、少なくとももう 1 つセグメントが必要です。URN の例は次のようになります。

urn:localhost:service:StoreService

この URN の 4 つ目のセグメントの値は `StoreService` なので、結果的に次のような URL になります。

http://localhost:8080/active-bpel/services/StoreService/

### URN マッピングの管理

実行できる操作は次のとおりです。

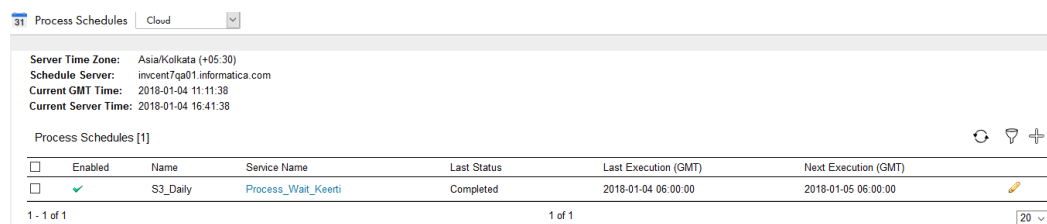
- 新しい URN を追加し、[新しい URN マッピングの追加] セクションの URN フィールドと URL フィールドに入力して、[マッピングの追加/更新] ボタンを押します。
- 既存の URN を編集し、このタブの上部にあるリンクをクリックします。[新しい URN マッピングの追加] セクションの URN フィールドと URL フィールドは、プロセスサーバーによって自動的に入力されます。変更を加えたら、[マッピングの追加/更新] ボタンを押します。
- マッピングを削除するには、マッピングの横にあるチェックボックスを選択し、[削除] を選択します。

## 第 4 章

# プロセススケジュール

ルーチンのメンテナンスタスクまたはその他の定期タスクを実行するために、BPEL プロセスが作成される場合があります。必要であれば、このプロセスをデプロイし、実行するスケジュールを毎週や月の特定の日などに設定出来ます。スケジュールは必要に応じて細かく設定出来ます（日中、秒単位）。

次の図は、[プロセススケジュール] ページを示しています。



次の表に、このページに表示される情報を示します。

フィールド	説明
有効	スケジュール名の隣のチェックマークは、プロセスが実行対象として有効になっていることを示します。
名前	スケジュールを説明する名前。例えば、プロセス名を使用し、その他の内容が分かる詳細を付けたりします。
サービス名	<b>【サービス定義】</b> リストからのサービス。サービス名を選択すると、サービス定義が表示されます。ヘルプにマウスカーソルを重ねると、サービスに関連するプロセス名が表示されます。
最終ステータス	アクティブなプロセスの実行状態。完了や実行中など。ステータスを選択すると、[アクティブなプロセスの詳細] ページが表示されます。ヘルプにマウスカーソルを重ねると、プロセス ID が表示されます。
最終実行	プロセスの最終実行日時（ある場合）
次回実行	スケジュールが有効になっている場合はスケジュールされた次回の実行日時。

クラスタ化された環境では、いつでもノードのいずれかがスケジュールマスターとして動作します。スケジュールされたプロセスが選定され、そのスケジュールマスターノードでのみ実行されます。スケジュールノードは最初に起動したサーバーに基づいて選定されます。Secure Agent クラスタ通信がドロップした場合にのみ、その他のノードがスケジュールマスタとして選定されます。また、プロセスサーバーはスケジュールされたプロセスの実行履歴を保持しません。

スケジュールされたプロセスを作成するには、ページの上部にある **【スケジュールの作成】** リンクをクリックします。プロセスサーバーは、「**新しいプロセススケジュールの作成**」で説明されているダイアログボックスを表示します。

「**実行スケジュールの管理および変更**」も参照して下さい。

## 新しいプロセススケジュールの作成

スケジュールされたプロセスを作成するには、**【プロセススケジュール】** ページの一番上にある、**【追加】** をクリックします。次のようなダイアログボックスが表示されます。

Execution Schedule

Execution Schedule

Name:

Untitled Schedule

Frequency:

Once

Start Date (GMT):

2018-01-04

(yyyy-mm-dd)

Start Time (GMT):

00:00:00

(hh:mm:ss)

Run Options:

☒ Skip if running

☐ Terminate if running

☐ Create new process

Service Type:

☒ BPEL Service

☐ IPD Process

Service Name:

0CBfPxmws9sdKkH9w443Sk/AgentLookupInvoiceDetails

Input Document:

?

OK

Cancel

実行スケジュール	
名前	スケジュールリストに表示する名前を指定します。
頻度	<div><div>- <b>【1 回】</b>。指定した開始日と時間に実行するには、これを選択します。</div><div><b>ヒント:</b> このオプションは、テストを実行する場合に適しています。その後、さまざまな実行スケジュールの詳細を複製できます。</div><div>- <b>【日中】</b>。1 日に複数回実行します。</div><div>- <b>【毎日】</b>、<b>【毎週】</b>、または <b>【毎月】</b> を使用すると、スケジュールの頻度をさらに細かく管理できます。</div></div>
サーバー上の開始日	1 回のみの実行（頻度=1 回）の場合は、日付を入力するか、 <b>【日付選択ツール】</b> を選択して日付を入力します。 <b>【日付選択ツール】</b> で、 <b>【今】</b> を選択して今日の日付を入力します。
サーバー上の開始時刻	デフォルトは深夜 0 時（00:00:00）です。時刻を入力するか、 <b>【日付選択ツール】</b> を選択して時刻を入力します。 <b>【日付選択ツール】</b> で、 <b>【今】</b> を選択して現在の時刻を入力します。

実行スケジュール	
スケジュールオプション	<ul style="list-style-type: none"> <li>- <b>【日中】</b> の場合は、時間数、分数、または秒数を選択します。オプションで、開始時間と終了時間を有効にします。</li> <li>- <b>【毎日】</b> の場合は、<b>【毎日】</b> (デフォルト)、<b>【すべての平日】</b> (月-金)、または <b>【n 日ごと】</b> を選択します。</li> <li>- <b>【毎週】</b> の場合、1 日以上の曜日を選択します。例えば、毎週月曜日に実行するには、<b>【月曜日】</b> を選択します。毎週月曜日と木曜日に実行するには、<b>【月曜日】</b> と <b>【木曜日】</b> を選択します。オプションで、何週おきに実行するかを 1 から n の範囲で指定できます。n は、2 週目、3 週目、n 週目など何番目の週にスケジュールするかを示します (例: 12 週目)。</li> <li>- <b>【毎月】</b> の場合は、特定の日付を選択するか、月の 1 番目、2 番目、3 番目、4 番目、または最後の日を選択します。例えば、月の最後の月曜日に実行するには、<b>【最終月曜日】</b> を選択します。<b>ヒント:</b> 毎月最終日に実行するには、日付として 31 を入力します。正しい最終日に調整されます。個別に月を選択することも可能です。デフォルトでは、すべての月が選択されています。</li> </ul>
実行時オプション	<p>次のオプションから選択します。</p> <ul style="list-style-type: none"> <li>- <b>前回のインスタンスがまだ実行中の場合はスキップする</b> (デフォルト)。前回のインスタンスが完了していない場合、他のプロセスインスタンスはスケジュール設定しません。</li> <li>- <b>前回のインスタンスがまだ実行中の場合でも終了する</b>。新しいインスタンスを開始する前に、前回スケジュール設定された実行を終了します。</li> <li>- <b>常に新しいプロセスを作成する</b>。以前の実行の状態に関係なく、新しいプロセス実行をスケジュール設定します。</li> </ul>
サービスタイプ	サービスが BPEL か IPD ベースかを選択します。
サービス名/ サービスの選択	<p>サービスタイプが BPEL の場合は、サービス名を入力します。<b>【サービス定義】</b> リストを開いて名前を表示し、サービス名を (拡張子なしで) 入力します (myScheduledService など)。</p> <p>サービスタイプが IPD の場合は、選択リストからいずれかのサービスを選択します。</p>
入力ドキュメント/ POST 本文	<p>プロセスが Process Developer を使用して実装され、BPEL タイプの場合は、次のいずれかの手順を実行します。</p> <ul style="list-style-type: none"> <li>- プロセスが SOAP インタフェースを実装する場合は、SOAP 本文要素内に含まれる SOAP 入力メッセージデータ (プロセスを呼び出すために使用される XML 入力メッセージ) を入力します。データは、プロセスの WSDL インタフェースに準拠している必要があります。</li> <li>- プロセスが REST インタフェースを実装する場合は、まず、REST サービスの SOAP ラッパーを作成する必要があります。続いて、上に示す指示に従います。</li> </ul> <p>Process Designer を使用してプロセスを実装し、プロセスが Informatica Process Definition (IPD) タイプである場合は、プロセスを呼び出すために使用される POST メッセージの JSON 本文を入力する必要があります。</p>

この情報の入力後、プロセスサーバーはスケジュール設定された新しいプロセスを作成し、それが [スケジュール済みプロセス] ページに表示されます。

「実行スケジュールの管理および変更」も参照してください。

# 実行スケジュールの管理および変更

リストを管理するには、プロセススケジュールリストのアイコン、チェックボックス、およびドロップダウンリストを使用します。

フィールド	説明
鉛筆アイコン (編集)	鉛筆アイコンを選択すると、スケジュールを編集できます。
選択/すべて選択	個々の実行スケジュールを選択するか、[すべて選択] チェックボックスを選択して、スケジュールの実行、複製、削除、有効化、無効化を行います。
今すぐ実行	選択したプロセスを実行します。
複製	選択したスケジュールを複製します。初期状態では、新しいスケジュールは無効です。
削除	選択した実行スケジュールを、すべての詳細情報を含め削除します。
有効化/無効化	選択した実行スケジュールを有効化または無効化します。スケジュールを有効にすると、プロセスは次回の実行時に実行されます。 <b>ヒント:</b> 無効化されたスケジュールを即座に実行（今すぐ実行）、複製、または削除することもできます。

[「新しいプロセススケジュールの作成」 \(ページ 60\)](#)も参照してください。

## 第 5 章

# プロセス

**【プロセス】** ページには、組織内で実行中または実行されたプロセスインスタンスのリストとプロセスバージョンが併せて表示されます。状態は、[実行中]、[一時停止中]、[完了]、[補償可能]（サブプロセス用）、または[フォールト] になる場合があります。

検索機能を使用すると、アクティブなプロセスインスタンスを素早く見つけることができます。プロセス ID を入力すると、プロセスインスタンスが表示されます。

次の表に、さまざまなプロセス状態を示します。

プロセス状態	意味
補償可能	サブプロセスが完了し、補償の対象となっています。
完了	正常に完了しています。
フォールト	プロセスはフォールトで終了しています。
実行中	プロセスは正常に実行中です。
一時停止中	何らかの理由でプロセスが一時停止しています。

実行中、一時停止中、フォールト、完了のプロセスのグループを選択し、それらに対してアクションを実行できます。

**【プロセス】** ページでは、次のタスクを実行できます。

1. 操作するプロセスのリストを表示します。選択フィルタを使用するとカスタマイズしたリストを作成できます。
2. 現在のページに表示されている一部または全部のプロセスを選択します。  
各ページに表示される最大レコード数まで選択できます。後続のページでプロセスは選択できません。各ページに表示するレコードの数を設定できます。
3. **【アクション】** リストから、次のいずれかのアクションを実行します
  - 終了: 実行中のプロセスを終了します。
  - 一時停止: 実行中のプロセスを一時停止します。
  - 再開: 一時停止したプロセスを再開します。

**注:** プロセスを再開する際、アプリケーション統合では、プロセスを再開して更新されたプロセスのステータスを表示するのに数秒かかります。プロセスを再開している間にプロセスのステータスをチェックした場合、プロセスのステータスは **【一時停止中】** と表示されます。このステータスは無視してください。プロセスが再開され、実行を開始した後にプロセスのステータスが更新されます。

- 再開: 完了したプロセス、フォールトになったプロセス、または一時停止したプロセスを再開します。詳細については、「[プロセスの再開](#)」(ページ 95)を参照してください。
- 削除: フォールトになったプロセスまたは完了したプロセスを削除します。
- 再試行: 一時停止中（フォールト）のプロセスを再試行します。プロセスを再試行する場合、プロセスはフォールトになったアクティビティを再開し、再試行します。

**注:** プロセスを再試行してすぐに再開すると、次のエラーが発生します。

The required logs are not available for the process instance <Instance ID> for the Tenant Context <Org ID>

再試行アクションが完了するまで待ってから、プロセスを再開してください。

**【プロセス】** ページからは、次のページにアクセスできます。

#### プロセスビューの詳細

プロセス ID をクリックすると、**【プロセスビューの詳細】** ページが表示されます。プロセスアクティビティの時系列リスト、デザイナに表示される状態のプロセス、詳細なステップ情報などを参照できます。**【アクティブなプロセスの詳細】** ページにアクセスすることもできます。

#### プロセスバージョンのリスト

プロセス名をクリックすると、**【プロセスバージョンのリスト】** ページが表示されます。ターゲット名前空間、バージョン名、バージョン番号、プラン ID、オンラインの日付、オフラインの日付、デプロイされた日付などを参照できます。

#### デプロイされたプロセスバージョンの詳細

プロセスのバージョン番号をクリックすると、**【デプロイされたプロセスバージョンの詳細】** ページが表示されます。このページでは、プロセスバージョンのライフサイクルを参照する、プロセスバージョンを更新する、マイロールやパートナーロールの詳細を参照すること以外にも、さまざまな処理を行えます。

## プロセスフィルタ

**【プロセス】** ページには、組織内で実行中または実行されたプロセスインスタンスのリストが表示されます。フィルタを適用することで、監視と分析が必要なプロセスを表示できます。

次のタイプのプロセスフィルタを作成できます。

#### 簡易プロセスフィルタ

簡易プロセスフィルタを作成して、フィールド、演算子、値を基にプロセスをフィルタリングすることができます。

#### 詳細プロセスフィルタ

クエリを使用して、詳細プロセスフィルタ条件を定義できます。定義済みリストの幅広いプロパティを使用し、論理演算子で式をグループ化することでクエリを作成します。

## 簡易プロセスフィルタ

簡易プロセスフィルタを作成して、フィールド、演算子、値を基にプロセスをフィルタリングすることができます。

1 つ以上の簡易プロセスフィルタを作成できます。複数の簡易プロセスフィルタを作成すると、プロセスサーバーによって AND 演算子ですべてのフィルタが結合され、全フィルタ条件を満たすプロセスが表示されます。

例えば次の図に示すように、今日完了したプロセスを表示する場合は、【ステータス】フィールドを【完了】に、【終了日】フィールドを【今日】に設定した、2つの簡易プロセスフィルタ条件を作成できます。

Processes (500) Updated 7:20 PM IST / 1:50 PM UTC Find

Status: Completed End Date: Today Add Field -OR- Use a Query

ID	Name	Version	Start Date	End Date	Status
465766064211775488	P-run-30s	1	2020/7/9 06:28:03:2...	2020/7/9 06:28...	Completed
465766064161443840	P-run-90s	1	2020/7/9 06:28:03:2...	2020/7/9 06:29...	Completed
465765812461260800	P-run-30s	1	2020/7/9 06:27:03:2...	2020/7/9 06:27...	Completed
465765812360597504	P-run-90s	1	2020/7/9 06:27:03:1...	2020/7/9 06:28...	Completed
465765560777854976	P-run-30s	1	2020/7/9 06:26:03:1...	2020/7/9 06:26...	Completed
465765560672997376	P-run-90s	1	2020/7/9 06:26:03:1...	2020/7/9 06:27...	Completed
465765309102837760	P-run-30s	1	2020/7/9 06:25:03:1...	2020/7/9 06:25...	Completed
465765308989591552	P-run-90s	1	2020/7/9 06:25:03:1...	2020/7/9 06:26...	Completed
465765057385877504	P-run-30s	1	2020/7/9 06:24:03:1...	2020/7/9 06:24...	Completed
465765057310380032	P-run-90s	1	2020/7/9 06:24:03:1...	2020/7/9 06:25...	Completed
465764805694083072	P-run-30s	1	2020/7/9 06:23:03:1...	2020/7/9 06:23...	Completed
465764805585031168	P-run-90s	1	2020/7/9 06:23:03:1...	2020/7/9 06:24...	Completed

1 - 25 of 500 < 1 of 20 > Items per Page: 25

## 簡易プロセスフィルタの設定

- アプリケーション統合コンソールで、左側のナビゲーションバーにある【プロセス】をクリックします。
- 右のペインで【フィルタ】アイコンをクリックします。

【フィールドの追加】オプションと【クエリの使用】オプションが表示されます。

- 【フィールドの追加】の横にある下矢印をクリックし、プロセスをフィルタリングするための基準とする次のフィールドのいずれかを選択します。

- ID
- 名前
- バージョン
- 開始日
- 終了日
- グループ
- ステータス

次の図は、プロセスをフィルタリングする基準に指定できるフィールドを示しています。

Processes (500) Updated 7:27 PM IST / 1:57 PM UTC Find

Add Field -OR- Use a Query

ID	Name	Version	Start Date	End Date	Status
465766064211775488	P-run-30s	1	2020/7/9 06:28:03:2...	2020/7/9 06:28...	Completed
465766064161443840	P-run-90s	1	2020/7/9 06:28:03:2...	2020/7/9 06:29...	Completed
465765812461260800	P-run-30s	1	2020/7/9 06:27:03:2...	2020/7/9 06:27...	Completed
465765812360597504	P-run-90s	1	2020/7/9 06:27:03:1...	2020/7/9 06:28...	Completed
465765560777854976	P-run-30s	1	2020/7/9 06:26:03:1...	2020/7/9 06:26...	Completed
465765560672997376	P-run-90s	1	2020/7/9 06:26:03:1...	2020/7/9 06:27...	Completed
465765309102837760	P-run-30s	1	2020/7/9 06:25:03:1...	2020/7/9 06:25...	Completed
465765308989591552	P-run-90s	1	2020/7/9 06:25:03:1...	2020/7/9 06:26...	Completed
465765057385877504	P-run-30s	1	2020/7/9 06:24:03:1...	2020/7/9 06:24...	Completed
465765057310380032	P-run-90s	1	2020/7/9 06:24:03:1...	2020/7/9 06:25...	Completed
465764805694083072	P-run-30s	1	2020/7/9 06:23:03:1...	2020/7/9 06:23...	Completed
465764805585031168	P-run-90s	1	2020/7/9 06:23:03:1...	2020/7/9 06:24...	Completed

1 - 25 of 500 < 1 of 20 > Items per Page: 25

4. 選択したフィールドを基にフィルタ条件を設定します。

次の表に、選択したフィールドを基に完了させる必要がある手順を示します。

フィールド	説明
ID	<p>プロセスのインスタンス ID。各プロセス実行には一意のインスタンス ID があります。</p> <p>[ID] を選択した場合は、次の手順を完了させます。</p> <ol style="list-style-type: none"> <li>次のいずれかの演算子を選択します。 <ul style="list-style-type: none"> <li>次の値に等しい</li> <li>次の値に等しくない</li> <li>次の値より小さい</li> <li>次の値より小さいか等しい</li> <li>次の値より大きい</li> <li>次の値以上</li> <li>期間</li> </ul> </li> <li>フィルタ条件の値を入力します。</li> </ol> <p>[Between] 演算子を選択した場合は、2 つの値を入力して、プロセスをフィルタリングするための基準とする範囲を定義します。</p>
バージョン	<p>プロセスのバージョン番号。</p> <p>[バージョン] を選択した場合は、次の手順を完了させます。</p> <ol style="list-style-type: none"> <li>次のいずれかの演算子を選択します。 <ul style="list-style-type: none"> <li>次の値に等しい</li> <li>次の値に等しくない</li> <li>次の値より小さい</li> <li>次の値より小さいか等しい</li> <li>次の値より大きい</li> <li>次の値以上</li> <li>期間</li> </ul> </li> <li>フィルタ条件の値を入力します。</li> </ol> <p>[Between] 演算子を選択した場合は、2 つの値を入力して、プロセスをフィルタリングするための基準とする範囲を定義します。</p>
名前	<p>プロセスの名前。</p> <p>[名前] を選択した場合は、フィルタ条件の値を入力します。</p>
グループ	<p>プロセスが属するグループ。グループ名はプロセスデプロイメント記述子 (PDD) で指定し、<b>[デプロイ済みプロセスバージョン詳細]</b> ページに表示されます。</p> <p>[グループ] を選択した場合は、フィルタ条件の値を入力します。</p>

フィールド	説明
開始日	<p>プロセス実行が開始した日付。</p> <p>【開始日】を選択した場合は、プロセスをフィルタリングするための基準とする、次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> <li>• 任意の時刻</li> <li>• 今日</li> <li>• 先週</li> <li>• 先月</li> <li>• 昨年</li> <li>• カスタム範囲</li> </ul> <p>【カスタム範囲】を選択した場合は、カレンダーをクリックしてフィルタ条件の日付と時間の範囲を指定します。</p>
終了日	<p>プロセスが完了、フォールト、または停止/一時停止した日付。</p> <p>【終了日】を選択した場合は、プロセスをフィルタリングするための基準とする、次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> <li>• 任意の時刻</li> <li>• 今日</li> <li>• 先週</li> <li>• 先月</li> <li>• 昨年</li> <li>• カスタム範囲</li> </ul> <p>【カスタム範囲】を選択した場合は、カレンダーをクリックしてフィルタ条件の日付と時間の範囲を指定します。</p>
ステータス	<p>プロセス実行の状態。</p> <p>【ステータス】を選択した場合は、プロセスをフィルタリングするための基準とする、次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> <li>• 完了</li> <li>• フォールト</li> <li>• 実行中</li> <li>• 一時停止中</li> </ul>

5. 必要に応じて追加のフィルタを作成します。

プロセスサーバーは AND 演算子ですべてのフィルタを結合し、全フィルタ条件を満たすプロセスを表示します。

フィルタを削除するには、フィルタ条件の横にある【閉じる】をクリックします。すべてのフィルタを削除し、すべてのプロセスを表示するには、【フィルタ】アイコンの横にある【閉じる】をクリックします。

## 詳細プロセスフィルタ

クエリを作成して、詳細プロセスフィルタ条件を定義できます。定義済みリストの幅広いプロパティを使用し、論理演算子で式をグループ化することでクエリを作成することができます。

次の図は、【クエリビルダ】ページを示しています。

Query Builder

×

Double-click the entries in the lists to insert the entries into the query. After you click Apply, all filters that were previously applied will be removed.

Process Properties

End Date  
Engine ID  
Group  
ID  
Initiator


Property Codes

▶ State  
▶ State Reason

Operators

and (logical AND)  
or (logical OR)  
= (equal to)  
!= (not equal to)  
> (greater than)

Insert Date



Query:

Clear

?

Apply

Cancel

【プロセスのプロパティ】リストと【プロパティコード】リストを使用して、プロセスをフィルタリングするための基準とするプロパティを選択できます。クエリに含める演算子を定義するには、【演算子】リストを使用します。

次の表に、**【プロセスのプロパティ】** リストで利用可能なプロパティを示します。

プロセスのプロパティ	説明	クエリの例
終了日	<p>プロセスが完了、フォールト、または停止/一時停止した日付。</p> <p><b>【終了日】</b> フィールドでは協定世界時 (UTC) が使用されます。</p> <p>ダブルクリックして  <code>[getProcessProperty("EndDate")]</code> プロパティをクエリに追加します。演算子を選択し、<b>【日付の挿入】</b> フィールドの下のカレンダーをクリックして終了日を選択します。</p> <p>クエリで <code>[=]</code> 演算子または <code>[!=]</code> 演算子を終了日と併せて使用する場合は、終了日をミリ秒で指定する必要があります。そうしないと、誤った結果が返されます。</p> <p>クエリで日付の範囲を使用して、終了日を基にプロセスをフィルタリングすることもできます。</p>	<p>次のいずれかのクエリを使用して、終了日が 2020 年 8 月 21 日であるプロセスをフィルタリングします。</p> <ul style="list-style-type: none"> <li><code>- getProcessProperty("EndDate") = "1597994701818"</code></li> <li><code>- getProcessProperty("EndDate") &gt; "2020/08/20" and getProcessProperty("EndDate") &lt; "2020/08/22"</code></li> </ul>
エンジン ID	<p>プロセスを実行するエンジンの ID。</p> <p>ダブルクリックして  <code>[getProcessProperty("EngineId")]</code> プロパティをクエリに追加します。演算子を選択し、エンジン ID の値を入力します。</p>	<p>次のクエリを使用して、エンジン ID が 1 であるプロセスをフィルタリングします。</p> <pre>getProcessProperty("EngineId") = 1</pre>
グループ	<p>プロセスが属するグループ。グループ名はプロセスデプロイメント記述子 (PDD) で指定し、<b>【デプロイ済みプロセスバージョン詳細】</b> ページに表示されます。</p> <p>ダブルクリックして <code>[getProcessProperty("Group")]</code> プロパティをクエリに追加します。演算子を選択し、グループ名の値を入力します。</p> <p>グループ名を単一引用符 (') または二重引用符 (") で囲みます。</p>	<p>次のクエリを使用して、North West Region グループに属するプロセスをフィルタリングします。</p> <pre>getProcessProperty("Group") = "North West Region"</pre>
ID	<p>プロセスのインスタンス ID。各プロセス実行には一意のインスタンス ID があります。</p> <p>ダブルクリックして <code>[getProcessProperty("Id")]</code> プロパティをクエリに追加します。演算子を選択し、ID の値を入力します。</p>	<p>次のクエリを使用して、インスタンス ID が 434251505437396992 より大きいプロセスをフィルタリングします。</p> <pre>getProcessProperty("Id") &gt; 434251505437396992</pre>
イニシエータ	<p>ワークフローを開始したユーザーの名前。</p> <p>ダブルクリックして <code>[getProcessProperty("Initiator")]</code> プロパティをクエリに追加します。演算子を選択し、イニシエータの名前を入力します。</p> <p>イニシエータ名を単一引用符 (') または二重引用符 (") で囲みます。</p>	<p>次のクエリを使用して、John が開始したプロセスをフィルタリングします。</p> <pre>getProcessProperty("Initiator") = 'John'</pre>

プロセスのプロパティ	説明	クエリの例
名前	<p>プロセスの名前。</p> <p>ダブルクリックして <code>getProcessProperty("Name")</code> プロパティをクエリに追加します。演算子を選択し、プロセス名を入力します。</p> <p>プロセス名を単一引用符 (') または二重引用符 (") で囲みます。</p>	<p>次のクエリを使用して、名前が <code>LoanApproval</code> であるプロセスをフィルタリングします。</p> <pre>getProcessProperty("Name") = "LoanApproval"</pre>
名前空間	<p>プロセスのターゲット名前空間。</p> <p>ターゲット名前空間には次の構文を使用します。</p> <pre>urn:screenflow:process:&lt;process_name&gt;</pre> <p>ターゲット名前空間の値はプロセスごとに一意です。</p> <p>ダブルクリックして <code>getProcessProperty("Namespace")</code> プロパティをクエリに追加します。演算子を選択し、ターゲット名前空間を入力します。</p> <p>ターゲット名前空間を単一引用符 (') または二重引用符 (") で囲みます。</p>	<p>次のクエリを使用して、ターゲット名前空間が <code>urn:screenflow:process:LoanApproval</code> であるプロセスをフィルタリングします。</p> <pre>getProcessProperty("Namespace") = 'urn:screenflow:process:LoanApproval'</pre>
開始日	<p>プロセス実行が開始した日付。</p> <p>【開始日】フィールドでは協定世界時 (UTC) が使用されます。</p> <p>ダブルクリックして <code>getProcessProperty("StartDate")</code> プロパティをクエリに追加します。演算子を選択し、【日付の挿入】フィールドの下のカレンダーをクリックして開始日を選択します。</p> <p>クエリで <code>[=]</code> 演算子または <code>[!=]</code> 演算子を開始日と併せて使用する場合は、開始日をミリ秒で指定する必要があります。そうしないと、誤った結果が返されます。</p> <p>クエリで日付の範囲を使用して、開始日を基にプロセスをフィルタリングすることもできます。</p>	<p>次のいずれかのクエリを使用して、開始日が 2020 年 8 月 21 日であるプロセスをフィルタリングします。</p> <pre>- getProcessProperty("StartDate") = "1597994701818" - getProcessProperty("StartDate") &gt; "2020/08/20" and getProcessProperty("StartDate") &lt; "2020/08/22"</pre>

プロセスのプロパティ	説明	クエリの例
状態	<p>プロセス実行の状態。</p> <p>ダブルクリックして <code>getProcessProperty("State")</code> プロパティをクエリに追加します。演算子を選択します。【プロパティコード】リストで【状態】リストを展開し、状態の値をダブルクリックして、関連する状態コードをクエリに挿入します。</p> <p>次のリストに、状態の値および関連する状態コードを示します。</p> <ul style="list-style-type: none"> <li>- 実行中 1: プロセスは実行中です。</li> <li>- 一時停止中 2: プロセスは一時停止中です。</li> <li>- 完了 3: プロセスは正常に完了しました。</li> <li>- フォールト 4: プロセスはフォールトで終了しています。</li> <li>- 補償可能 5: サブプロセスが完了し、補償の対象となっています。</li> </ul>	<p>次のクエリを使用して、フォールト状態のプロセスをフィルタリングします。</p> <pre>getProcessProperty("State") = "4"</pre>
状態の理由	<p>プロセスが一時停止になっている理由。</p> <p>ダブルクリックして <code>getProcessProperty("StateReason")</code> プロパティをクエリに追加します。演算子を選択します。【プロパティコード】リストで【状態の理由】リストを展開し、状態の理由の値をダブルクリックして、関連する状態の理由コードをクエリに挿入します。</p> <p>次のリストに、状態の理由の値および関連する状態の理由コードを示します。</p> <ul style="list-style-type: none"> <li>- 一時停止中（手動） 0: 手動で一時停止されたプロセスに適用されます。</li> <li>- 一時停止中（フォールト） 1: キャッチされなかったフォールトにより一時停止になったプロセスに適用されます。</li> <li>- 一時停止中（アクティビティ） 2: BPEL の一時停止アクティビティで一時停止になったプロセスに適用されます。</li> <li>- 一時停止中（移行エラー） 7: 新しいプロセスバージョンに移行されたプロセスの移行エラーにより一時停止になったプロセスに適用されます。</li> <li>- 一時停止中（移行警告） 8: 新しいプロセスバージョンに移行されたプロセスの移行警告により一時停止になったプロセスに適用されます。警告は、プロセス再開前に確認と更新が必要な可能性がある、新しいバージョンで見つかった相違点を示します。</li> <li>- 移行前コピー 20: 新しいプロセスインスタンスが再開または終了するまで一時的に保存されている古いプロセスインスタンスに適用されます。</li> </ul>	<p>次のクエリを使用して、フォールトになったために一時停止状態になっているプロセスをフィルタリングします。</p> <pre>getProcessProperty("StateReason") = 1</pre>

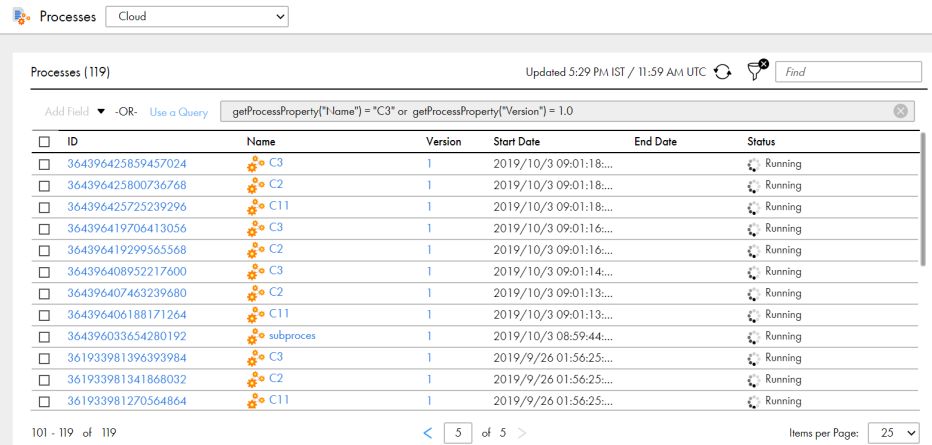
プロセスのプロパティ	説明	クエリの例
テナント ID	<p>プロセスのテナント識別番号。テナント識別番号は組織ごとに一意です。</p> <p>ダブルクリックして <code>【getProcessProperty("TenantId")】</code> プロパティをクエリに追加します。演算子を選択し、テナント ID の値を入力します。</p> <p>テナント ID を単一引用符 (') または二重引用符 (") で囲みます。</p>	<p>次のクエリを使用して、テナント ID が <code>3i5mW6m6816cHjNTPH8LwE</code> であるプロセスをフィルタリングします。</p> <pre>getProcessProperty("TenantId") = "3i5mW6m6816cHjNTPH8LwE"</pre>
タイトル	<p>プロセスの名前。</p> <p>ダブルクリックして <code>【getProcessProperty("Title")】</code> プロパティをクエリに追加します。演算子を選択し、プロセス名を入力します。</p> <p>プロセス名を単一引用符 (') または二重引用符 (") で囲みます。</p>	<p>次のクエリを使用して、名前が <code>LoanApproval</code> であるプロセスをフィルタリングします。</p> <pre>getProcessProperty("Name") = "LoanApproval"</pre>
バージョン	<p>プロセスのバージョン番号。</p> <p>ダブルクリックして <code>【getProcessProperty("Version")】</code> プロパティをクエリに追加します。演算子を選択し、バージョン番号の値を入力します。</p>	<p>次のクエリを使用して、バージョン番号が 1 に等しいプロセスをフィルタリングします。</p> <pre>getProcessProperty("Version") = 1</pre>

次の表に、**【プロパティコード】** リストで利用可能なプロパティを示します。

プロパティコード	説明
状態	<p>プロセス実行の状態。</p> <p><b>【状態】</b> リストを展開し、状態の値をダブルクリックして、関連する状態コードをクエリに挿入します。</p> <p>次のリストに、状態の値および関連する状態コードを示します。</p> <ul style="list-style-type: none"> <li>- 実行中 1: プロセスは実行中です。</li> <li>- 一時停止中 2: プロセスは一時停止中です。</li> <li>- 完了 3: プロセスは正常に完了しました。</li> <li>- フォールト 4: プロセスはフォールトで終了しています。</li> <li>- 補償可能 5: サブプロセスが完了し、補償の対象となっています。</li> </ul>
状態の理由	<p>プロセスが一時停止になっている理由。</p> <p><b>【状態の理由】</b> リストを展開し、状態の理由の値をダブルクリックして、関連する状態の理由コードをクエリに挿入します。</p> <p>次のリストに、状態の理由の値および関連する状態の理由コードを示します。</p> <ul style="list-style-type: none"> <li>- 一時停止中（手動） 0: 手動で一時停止されたプロセスに適用されます。</li> <li>- 一時停止中（フォールト） 1: キャッチされなかったフォールトにより一時停止になったプロセスに適用されます。</li> <li>- 一時停止中（アクティビティ） 2: BPEL の一時停止アクティビティで一時停止になったプロセスに適用されます。</li> <li>- 一時停止中（移行エラー） 7: 新しいプロセスバージョンに移行されたプロセスの移行エラーにより一時停止になったプロセスに適用されます。</li> <li>- 一時停止中（移行警告） 8: 新しいプロセスバージョンに移行されたプロセスの移行警告により一時停止になったプロセスに適用されます。警告は、プロセス再開前に確認と更新が必要な可能性がある、新しいバージョンで見つかった相違点を示します。</li> <li>- 移行前コピー 20: 新しいプロセスインスタンスが再開または終了するまで一時的に保存されている古いプロセスインスタンスに適用されます。</li> </ul>

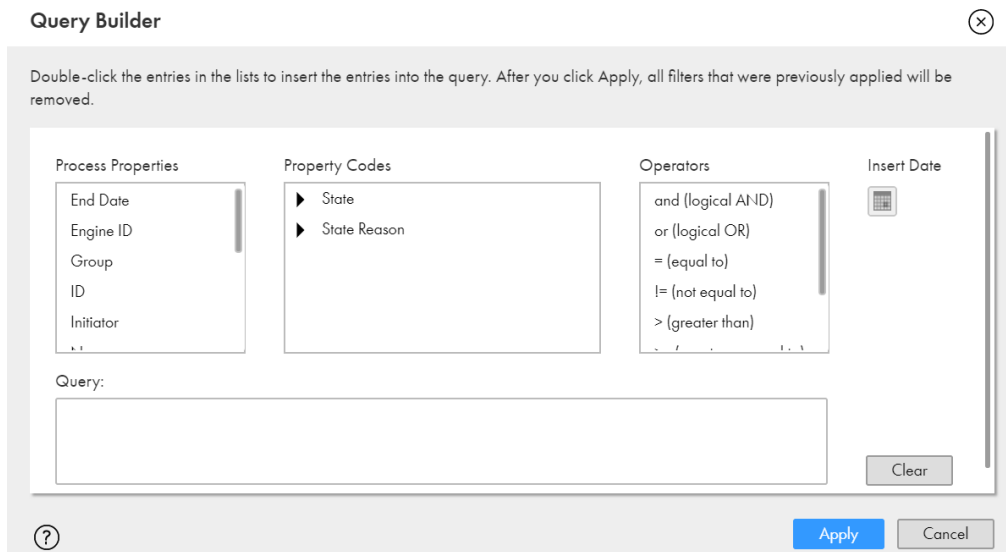
次の表に、クエリで使用できる演算子を示します。

演算子	説明
and	<p>論理積</p> <p>2つの条件を and 演算子で結合すると、プロセスサーバーは、両方の条件を満たすプロセスを表示します。</p> <p>例えば、次のようなクエリがあるとします。</p> <pre>getProcessProperty("Name") = "C3" and getProcessProperty("Version") = 1</pre> <p>次の図に示すように、このクエリによって、名前が C3 で、バージョン番号が 1 のプロセスが返されます。</p>  <p>注: and 演算子は、大文字と小文字を区別します。</p>
or	<p>論理和</p> <p>2つの条件を or 演算子で結合すると、プロセスサーバーは、2つの条件のいずれかを満たすプロセスを表示します。</p> <p>例えば、次のようなクエリがあるとします。</p> <pre>getProcessProperty("Name") = "C3" or getProcessProperty("Version") = 1</pre> <p>次の図に示すように、このクエリによって、名前が C3 のプロセスが返されます。</p>  <p>次の図に示すように、このクエリによって、バージョン番号が 1 のプロセスも返されます。</p>

演算子	説明
	 <p>注: or 演算子は、大文字と小文字を区別します。</p>
=	次の値に等しい
!=	次の値に等しくない
>	より大きい
>=	以上
<	より小さい
<=	以下

## 詳細プロセスフィルタの設定

- アプリケーション統合コンソールで、左側のナビゲーションバーにある【プロセス】をクリックします。
- 右のペインで【フィルタ】アイコンをクリックします。  
【フィールドの追加】オプションと【クエリの使用】オプションが表示されます。
- 【クエリの使用】をクリックします。  
【クエリビルダ】ページが表示されます。  
次の図は、【クエリビルダ】ページを示しています。



4. 【プロセスのプロパティ】、【プロパティコード】、【演算子】の各リストのエントリをダブルクリックして、それらのエントリをクエリに挿入します。任意でクエリを編集するか、【クリア】をクリックしてクエリをクリアします。

設定したクエリは【クエリ】ボックスに表示されます。

5. 【適用】をクリックします。

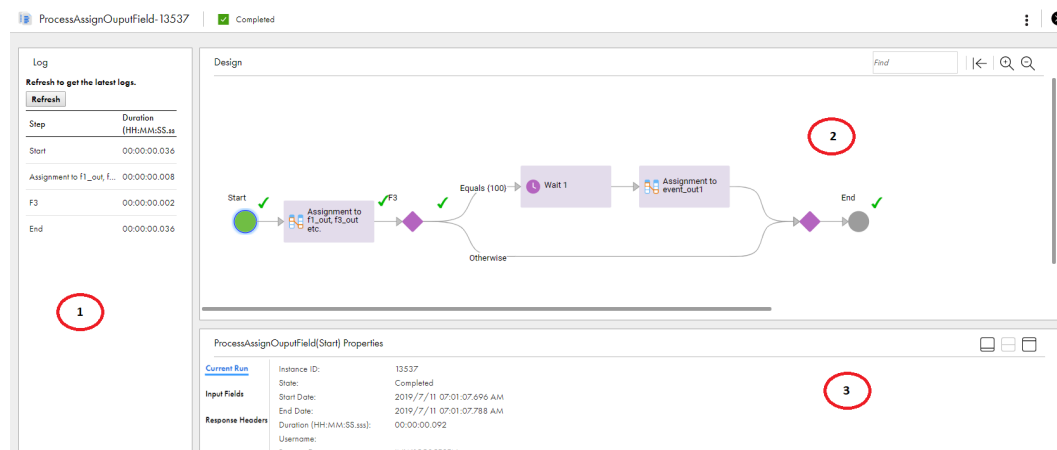
プロセスサーバーは、これまでに適用されたすべてのフィルタを削除し、作成済みクエリを満たすプロセスを表示します。

フィルタを削除し、すべてのプロセスを表示するには、フィルタ条件の横にある【閉じる】をクリックするか、【フィルタ】アイコンの横にある【閉じる】をクリックします。

## 【プロセスビューの詳細】 ページ

【プロセス】ページのプロセス ID をクリックすると、【プロセスビューの詳細】ページが表示されます。

次の図は、アプリケーションの統合を使用して作成されたプロセスの【プロセスビューの詳細】ページの例を示しています。



【プロセスビューの詳細】 ページには次のセクションが含まれます。

- セクション 1 には【ログ】 エリアが表示されます。プロセスアクティビティの時系列リストが表示されます。【ログ】 リストを検索してステップをクリックすると、アプリケーション統合コンソールは右側の対応するステップを強調表示します。
- セクション 2 には、プロセスが Process Designer の見え方で表示されます。
- セクション 3 には、強調表示されたステップに関する情報が表示されます。選択されたステップに応じて、このセクションにはフォールトの情報、その他の情報が表示されるか、何も表示されないことがあります。右上隅にある【詳細ビュー】 をクリックすると、このプロセスの【アクティブなプロセスの詳細】 ページが表示されます。

**注:** プロセスの割り当てステップのランタイム情報を表示するには、次のいずれかのオプションを使用します。

- 要求の HTTP ヘッダーとして X-Debug=true を設定する。
- 【次を使用して実行】 オプションを使用してプロセス入力を作成し、1 つ以上のプロセス入力でプロセスを実行する。

## 開始ステップのプロパティ

開始ステップのプロパティには、プロセスの現在の実行に関する情報が表示されます。

開始ステップをクリックすると、プロパティパネルに次の情報が表示されます。

プロパティ	説明
今回の実行	次のプロパティが表示されます。 <ul style="list-style-type: none"><li>- インスタンス ID。現在のプロセスインスタンスを識別する一意の番号。</li><li>- ステータス。プロセスインスタンスの状態。プロセスは、実行中、完了、フォールト、または一時停止中の状態となることがあります。</li><li>- 開始日。プロセスインスタンスが開始された日時。</li><li>- 終了日。プロセスインスタンスが終了した日時。プロセスが完了状態またはフォールト状態の場合は、終了日の値が表示されます。</li><li>- 期間。プロセスインスタンスが完了するまでに実行された時間。プロセスが完了状態またはフォールト状態の場合は、期間の値が表示されます。</li><li>- ユーザー名。プロセスのこのインスタンスを実行したユーザーの名前。</li><li>- ランタイム環境。プロセスが実行されたランタイム環境。</li></ul>
入力フィールド	プロセスで設定した入力フィールドの名前と値が表示されます。
応答ヘッダー	プロセスで設定した応答ヘッダーのヘッダー名と応答の値が表示されます。

## フォールトの詳細

プロセスでフォールトが発生すると、【開始】 ステップにプロセス全体の全体的なフォールト情報が表示され、【プロセスビューの詳細】 ページに、フォールトが発生したステップのフォールト情報が表示されます。

フォールト情報を表示するには、フォールトが発生したステップをクリックし、【フォールト】 タブをクリックします。【フォールト】 タブに、フォールトのコード、理由、ソース、詳細など、フォールトの解決に役立つ情報が表示されます。

次の表に、【フォールト】タブに表示されるフォールト情報を示します。

明細	説明
コード	フォールトを分類するコード。次のようなコードがあります。 <ul style="list-style-type: none"> <li>- AeAgentTunnelHttpException</li> <li>- AeInvokePrepareException</li> <li>- CreateDataException</li> <li>- DataProviderException</li> <li>- faultResponse</li> <li>- HTTP_500</li> <li>- Invalid Data</li> <li>- runtimeError</li> <li>- subLanguageExecutionFault</li> </ul>
理由	プロセスがあるステップでフォールトになった理由。
ソース	エラーが発生した BPEL ステップ。 <b>注:</b> フォールト処理を設定したステップでは、【ソース】フィールドは利用できません。代わりに【詳細】フィールドが表示されて、そのステップに対して設定したフォールト処理の詳細が示されます。
詳細	そのステップに対して設定したフォールト処理の詳細。

次の図は、ディシジョンステップでフォールトが発生したプロセスの【フォールト】タブを示しています。

ProcessFaultDecision-321931056414875648 ✖ Faulted

Log

Refresh to get the latest logs.

Refresh

Step	Duration (HH:MM:SS.ss)
Start	00:00:00.002
Assignment to textip, In...	00:00:00.001
Current User	00:00:00.003

Design

```

graph LR
    Start((Start)) --> Assign[Assignment to textip, Intip]
    Assign --> Decision{Current User}
    Decision -- "Equals {util:parseXML...}" --> Wait1[Wait 1]
    Decision -- "Otherwise" --> Wait2[Wait 2]
    Wait1 --> Join(( ))
    Wait2 --> Join
    Join --> End((End))
  
```

Current User Properties

**Fault**

Name	Value
Code	subLanguageExecutionFault
Reason	Unexpected exception found.
Source	/process/flow/scope[@name="Current_User"]

次の図は、フォールト処理を使用するスローステップの【フォールト】タブを示しています。

Check Year - with single fault-335363553454198784 | Failed

Log

Refresh to get the latest logs.

Refresh

Step	Duration (HH:MM:SS.ss)
Start	00:00:00.004
Temp Assignment	00:00:00.009
Assignment to MakesC...	00:00:00.001
Fault Process	00:00:00.037
Process two	00:00:01.264
Current Year	
Fault handler for process	00:00:00.003

Design

Fault handler for process Properties

Fault	
Name	Value
Code	505
Reason	Count has not been incremented
Details	Count has not been incremented. Throwing exception.

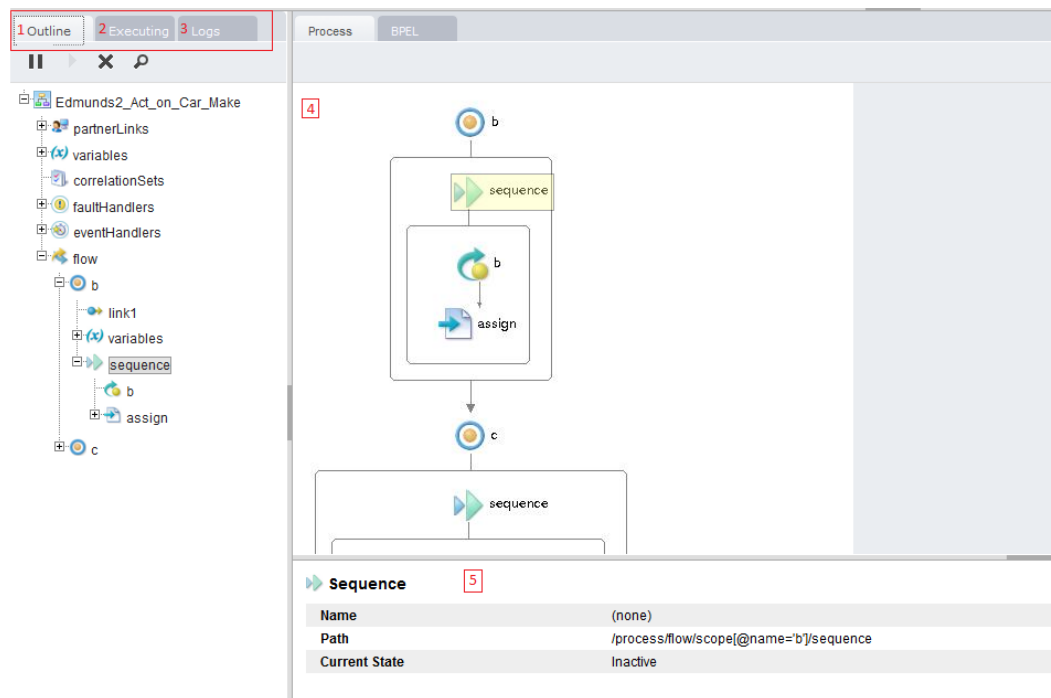
【ソース】フィールドの代わりに【詳細】フィールドが表示されていることに注目してください。このフィールドには、スローステップに対して設定したフォールト処理の詳細が表示されます。

## [アクティブなプロセスの詳細] ページ

ページでプロセスインスタンスを選択すると、[アクティブなプロセスの詳細] ページが表示されます。プロセスインスタンスに関するさまざまな詳細情報が表示されるほか、プロセスを一時停止したり、プロセス例外管理を実行できます。

【プロセス】の右上隅にある【詳細ビュー】をクリックすると、プロセスの [アクティブなプロセスの詳細] ページが表示されます。

次の図は、ページとページ内の領域を示しています。



次のコンポーネントを使用できます。

	領域	内容
1	[アウトライン] ビュー	BPEL プロセスの構造上の要素と、各アクティビティの現在のプロセス実行状態が表示されます。要素を選択して、そのプロパティと値を確認できます。
2	[実行中] ビュー	実行中のプロセスの現在のプロセス実行状態が表示されます。現在実行中のアクティビティで停止しています。プロセスが完了すると、実行中ビューは空白になります。
3	[ログ] ビュー	データの変更、フォールト、その他の詳細を分析するために使用できる実行の詳細が表示されます。プロセスを一時停止し、アクティビティのプロセス巻き戻しを実行することもできます。
4	[プロセス] ビュー	主なプロセスフローがキャンバスに表示されます。プロセスにイベント、フォールト、補償ハンドラがある場合は、対応するアクティビティを [アウトライン] から選択することでこれらを表示できます。例えば、Catch ハンドラを選択してすべてのフォールトハンドラを表示します。アクティビティを選択してそのプロパティを表示することもできます。BPEL タブには、プロセスのソース XML コードが表示されます。
5	[プロパティ] ビュー	他のコンポーネントのキャンバスから選択した選択済み要素について表示されます。例えば、[アウトライン] ビューから [プロセス] 要素が選択されると、現在の状態や開始日など、プロセスのプロパティと現在の値が表示されます。

[変数]、[相関セット]、[パートナーリンク]、および [スレッドのデバッグ] のツールバーから追加ウィンドウを開くと、データやステップの編集やアクティビティの再試行を簡単に行えます。

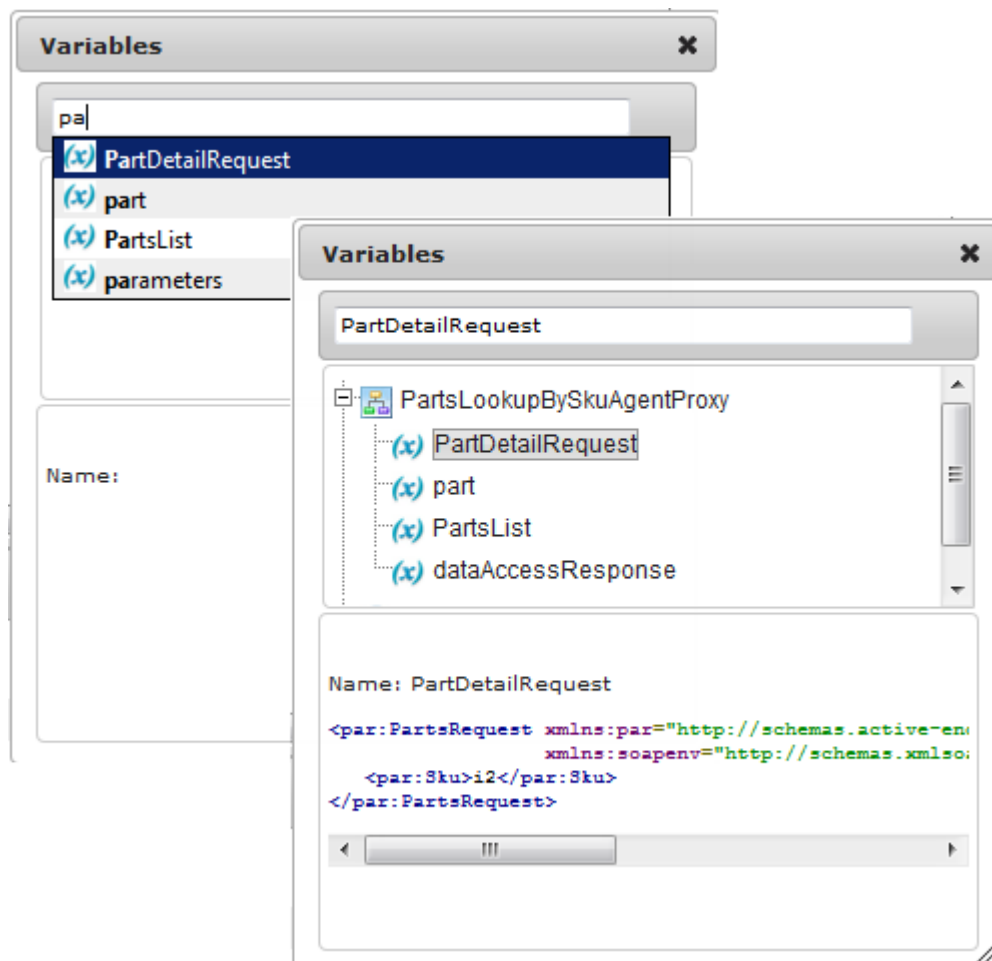
## ツールバー

ツールバーは、このページのさまざまなビューの選択を可能にするほか、プロセスの例外管理を実行する機能を備えています。



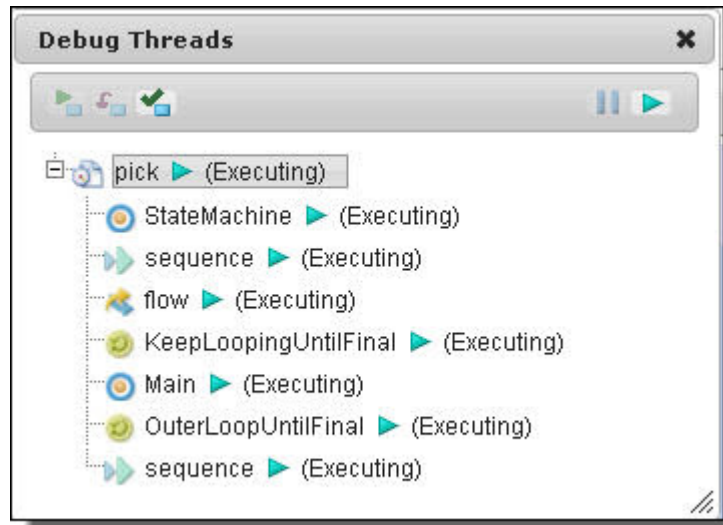
上の行	<p>上の行にある項目を使用すると、デフォルト（すべてのコンポーネント）、アウトライン（アウトライン+キャンバス）、プロパティ（プロパティ+キャンバス）、またはキャンバスのみのビューを選択できます。また、次の引数を指定した URL ビューパラメータを使用して、さまざまなビューを表示できます。</p> <ol style="list-style-type: none"> <li>1:（デフォルト）すべてのコンポーネントを表示（アウトライン、キャンバス、プロパティ）</li> <li>2: キャンバスのみ</li> <li>3: アウトラインおよびキャンバス</li> <li>4: キャンバスおよびプロパティ</li> </ol>
下の行	<p>いずれかのボタンを押すと、次のダイアログが表示されます。</p> <ol style="list-style-type: none"> <li>1. 変数</li> <li>2. パートナーリンク</li> <li>3. 相関セット</li> <li>4. スレッドのデバッグ</li> </ol>

[変数]、[パートナーリンク]、および [相関セット] ダイアログの動作は似ています。ここで示すのは [変数] ダイアログの例です。



変数の名前を入力する途中で、入力内容に合った変数が選択リストに表示されます（左のダイアログ）。いずれかを選択したら、ダイアログに変数の情報が表示されます。

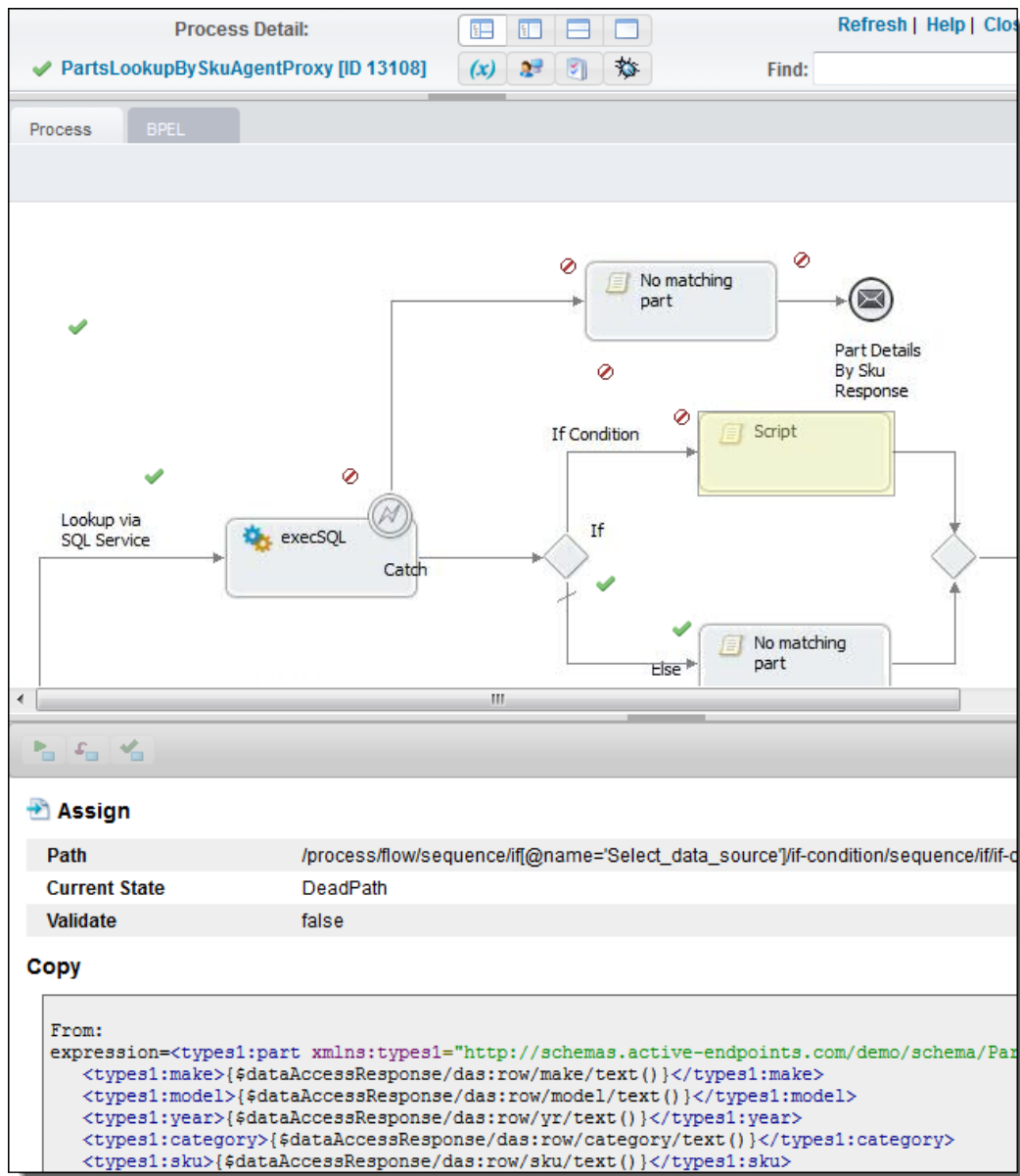
「スレッドのデバッグ」ダイアログを使用すると、プロセスを一時停止し、プロセスを順番に実行し、変数、パートナーリンク、または相関セットを編集できます。



## 「キャンバス」ビュー

「キャンバス」ビューには、メインプロセスフローとそのプロセスの実行パスが表示されます。「アウトライン」から選択すると、プロセスのフォールトハンドラ、イベントハンドラ、補償ハンドラも表示できます。例えば、Catchを選択するとすべてのフォールトハンドラが表示されます。

「キャンバス」ビューは、次の例のように表示されます。



詳細を表示するには、次の手順を実行します。

- 図からアクティビティを選択してプロパティを表示する。
- 図からアクティビティを選択して、アクティビティを「アウトライン」の表示対象とする。またはその逆をする。
- (Process Developer を使用している場合は) 人員アクティビティを選択して、アクティブなタスクの詳細を表示する。

図を印刷するには、右クリックメニューからブラウザの印刷オプションを選択します。図を印刷すると、「キャンバス」ビューに表示されているキャプションと同じキャプションが印刷されます。タイムスタンプは、ページが開かれた時刻または更新した時刻を表します。

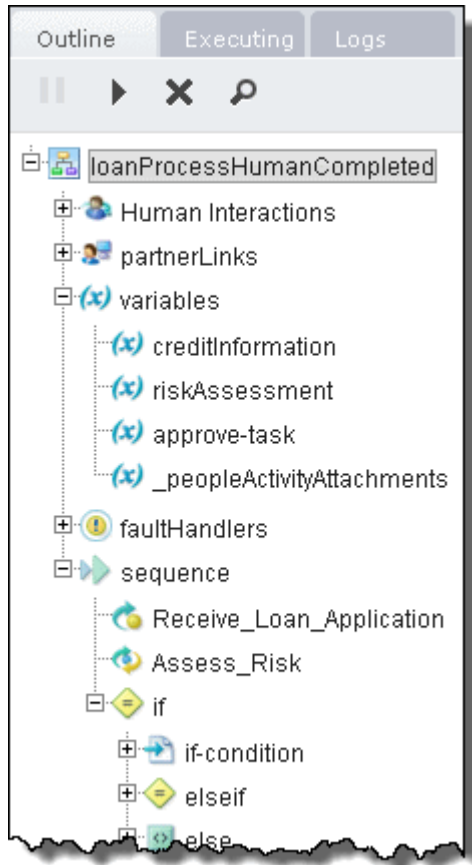
## 「実行中」ビュー

「実行中」ビューは、現在実行中のアクティビティのみを表示するフィルタです。このビューは、プロセスの例外管理を実行する実行ポイントを特定する際に便利です。

## [アウトライン] ビュー

[アウトライン] ビューを使用して、キャンバスで表示対象とするアクティビティを選択します。これらのアクティビティは必ずしも実行順に表示されるわけではありません。表示順はアクティビティの作成順と一致します。また、[アウトライン] 上のアイテムの一部はフローやシーケンスといったコンテナです。これらはキャンバスでは非表示になっていることがあります。フローを選択した場合は、フローの最初のアクティビティが表示対象となります。プラス記号が表示されている折りたたまれたコンテナを選択した場合は、コンテナ内のアクティビティのドリルダウンビューが表示されます。パートナーリンクまたは変数を選択した場合は、それらが [プロパティ] ビューに表示されます。

次の図は、プロセスインスタンスの [アウトライン] ビューを示しています。



### [アウトライン] ビューのメニューバー

[一時停止]、[再開]、[終了] ボタンは、プロセスが現在実行中か一時停止中の場合にのみ有効になります。これらの機能を使用すると、一時停止中のアクティビティまたは一時停止中（フォールト）のアクティビティの修正、再開、再試行、完了を行うことで、プロセスの例外管理を行えます。を参照してください。

### BPEL プロセスの構造要素

プロセス要素の詳細を表示するには、その要素を選択します。次の表に、各要素を示します。ここには、プロセスに存在する要素のみが表示されています。

【アウトライン】の プロセス要素	【プロパティ】に表示される詳細
アクティビティ	<p>【アウトライン】の【アクティビティ】セクションは、プロセス全体のメインコンテナを表すフロー（またはその他の）アクティビティから始まります。フローの中に、すべてのプロセスアクティビティのリストが表示されます。アクティビティは BPEL XML コードと同じ順序で表示されます。プロセスが Process Developer で設計されている場合、表示順は【アウトライン】ビューの順序と同じになります。</p> <p>【アクティビティ】リストは必ずしも実行順に表示されるわけではありません。各アクティビティに対して、実行状態とアクティビティの定義を確認することができます。</p>
関連セット	<p>これらの要素は、プロセスによって行われるさまざまな対話をトラッキングするメッセージプロパティ（定義と現在値）を表示します。関連セットにはメッセージプロパティが含まれ、各プロセス対話が一意に識別されるようにします。</p>
イベントハンドラ	<p>イベントハンドラはプロセススコープと同時に実行され、アラームや受信したメッセージに基づいてアクティビティを呼び出します。表示される情報は、イベントハンドリングアクティビティの名前、状態、詳細です。スコープは独自のローカルフォールトハンドラを持つことができます。</p>
フォールトハンドラ	<p>フォールトハンドラはプロセススコープと同時に実行され、アラームや受信したメッセージに基づいてアクティビティを呼び出します。これらのハンドラは、プロセスアクティビティによって投げられた Catch フォールトを表示します。これには、フォールドハンドリングアクティビティの名前、状態、詳細が含まれます。スコープは独自のローカルフォールトハンドラを持つことができます。</p>
ヒューマンインタラクション	<p>これは、ヒューマンタスクのための Informatica 拡張機能です。論理的な人員グループ、タスク、通知を表示します。</p>
リンク	<p>アクティビティがリンクのソースの場合、アクティビティノードの下にリンクが表示されます。リンクの状態（リンクが実行されたかどうか）、遷移状況（存在する場合）、リンクのターゲットアクティビティとなど、リンクのプロパティが表示されます。</p>
パートナーリンク	<p>パートナーリンクのタイプです。これは呼び出された Web サービスです。パートナーのロールや自分のロールが表示されます。</p> <p>これはパートナーリンクサービスのエンドポイント参照である場合があります。【プロパティ】ビューでアドレス情報を確認できます。スコープは独自のローカルパートナーリンクを持つことができます。</p>
プロセス名	<p>これは、プロセス修飾名のローカル部分です。プロセスインスタンスの現在の状態、開始/終了時刻、プロセスのデプロイメントの詳細が表示されます。場合によってはフォールトの詳細も表示されます。</p>

【アウトライン】の プロセス要素	【プロパティ】に表示される詳細
終了ハンドラ	プロセス内のアクティビティは、フォールドハンドリングを通して終了したり、並列の forEach の完了状態に基づいて早期終了したり、Process Developer 拡張機能のブレイクアクティビティを通して終了したりすることができます。
変数	<p>受信、操作したメッセージやその他のデータ、プロセスから送信されたメッセージやその他のデータが表示されます。変数には添付ファイルが含まれる場合があります。変数タイプについては、メッセージ、スキーマのタイプ、またはスキーマの要素が表示されます。</p> <p>変数の現在の値も表示されます。実行中のプロセスについては、値は【プロセスの詳細】ウィンドウを開いた時点あるいは更新した時点の現在値です。</p> <p>変数に添付ファイルがある場合、それらも表示され、ダウンロードできます。</p>

## プロセスの状態

プロセスには次の実行状態があります。

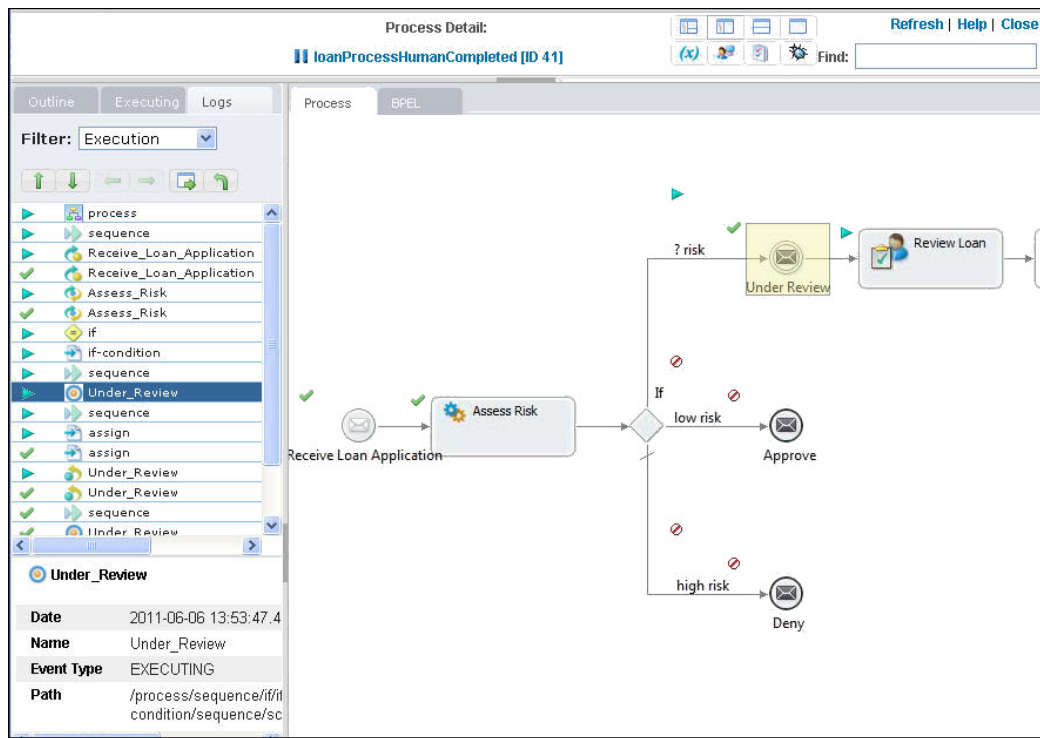
状態	意味
完了	正常に完了しています。
フォールト	フォールトまたは早期終了で終了しています。
実行中	【プロセスの詳細】ウィンドウを開いた時点の実行中プロセスのスナップショットです。プロセスは引き続き実行されますが、【プロセスの詳細】ウィンドウは、【更新】を選択するまで更新されません。
一時停止中（フォールト）	実行はフォールトになったアクティビティで停止しています。アクティビティに捕捉されていないフォールトがあり、プロセスは捕捉されていないフォールトで一時停止するように設定されています。
一時停止中（手動）	【プロセスの詳細】ウィンドウで【一時停止】が選択された時点でプロセスが停止しています。

プロセス移行の警告とエラーに該当する場合はその他の状態が表示されます。

## 【ログ】ビュー

【アクティブなプロセスの詳細】ページで、【ログ】タブを選択すると、実行アクティビティの時刻ベースのリストが表示されます。ログを使用して、データの変更やフォールトなどプロセスの実行イベントを分析できます。実行ログ内では While ループの反復など任意の時点に移動して、その実行ポイントでのプロセスの状態とデータを確認できます。イベントにステップスルーして、プロセスが時間の経過とともにどのように変化したかを表示できます。

実行ログを以下に示します。






実行ログは次のように使用します。

- データの変更、フォールト、アクティビティの実行状態、巻き戻しポイント、またはすべてを表示するフィルタを選択します。データの変更フィルタでは、コンテンツが変更された変数が表示されます。また、割り当てアクティビティで動的に変更されたパートナーリンクと論理人員グループも表示されます。
- イベントを選択し、[実行ポイントに移動]を選択してイベントをキャンバスの表示対象とします。プロセスの状態とデータは、選択したログイベントで作成されたスナップショットに基づきます。
- [実行ポイントに移動]を選択したら、ウィンドウの右上隅にある【現在に移動】を選択して、実行中のプロセスの現在実行中のアクティビティに戻ります。
- イベントを選択したら、[アウトライン]ビューに切り替え、選択した実行状態のプロセスのグラフを表示します。[アウトライン]では、その時点での変数を調べることもできます。

### アクティビティの状態

アクティビティの隣にあるアイコンで、各アクティビティの実行状態を判断できます。

アイコン	説明
	実行中
	実行準備完了
	完了
	フォールト。アクティビティの実行中にフォールトが投げられた場合に発生します。
	一時停止中（フォールト）。アクティビティの実行中にフォールトが投げられ、そのフォールトが捕捉されていない場合に発生します。必要に応じて、一時停止中（フォールト）のプロセスを修正するか再開することができます。

アイコン	説明
	終了。プロセスが手動で終了されたときに発生します。
	デッドパス
	一時停止中
(なし)	非アクティブ（アクティビティの初期状態）

実行中のプロセスについては、[プロセスの詳細] ウィンドウを更新すると、アクティビティの隣にあるアイコンが変化することがあります。

フォールト情報には次のものが含まれます。

列	説明
フォールト名	BPEL またはエンジンのフォールト名
フォールト名前空間	BPEL またはエンジンのフォールト名前空間
フォールトソース	フォールトを投げたプロセスアクティビティ
フォールトメッセージデータ	Throw または Catch フォールト変数のデータ
フォールト添付ファイル	変数の外部のコンテンツが含まれる場合があります、それらはダウンロード可能

フォールトについての詳細な情報は次の方法で確認できます。

- フォールトになったアクティビティを選択してフォールト名を表示する。
- フォールトになったアクティビティについては、データを修正したり、アクティビティを再試行または完了したりすることができます。

**ヒント:** [アウトライン] タブの **【ログの表示】**（虫めがねアイコン）を選択すると、プロセスログ全体をテキストファイルとして表示し、ダウンロードできます。

## 【プロセスバージョンのリスト】 ページ

次の図に、【プロセスバージョンのリスト】 ページの例を示します。

Lookup Invoice and Customer Details

Name:

Lookup\_Invoice\_and\_Customer\_Details-1

Target Namespace:

urn:screenflow:process:Lookup\_Invoice\_and\_Customer\_Details-1

Version	Plan Id	Online Date	Offline Date	Deployed Date	Processes	Migrated To	Status
18.0	144718805355429888	2018/02/04 08:20 AM		2018/02/04 08:20 AM	16		Online
16.0	144663975928758272	2018/02/04 04:42 AM		2018/02/04 04:43 AM	3		Offline

Set to Offline

プロパティ	説明
バージョン	新しいプロセスバージョンがデプロイされると、デプロイメント記述子でバージョン番号が指定されていない限り、バージョン番号が自動的に引き上げられます。番号の形式は N.nn で、N はメジャー、nn はマイナーです。
プラン ID	このバージョンに割り当てられた ID と、実行中のプロセスの関連ディスポジション。
オンラインの日付	オンライン（有効）の日付がデプロイメント記述子で指定されなかった場合、オンラインの日付はデプロイされた日付と同じになります。オンラインの日付が指定されていない限り、プロセスはデプロイされるとすぐにオンラインになります。
オフラインの日付	デプロイメント記述子で指定されていない限り、プロセスバージョンにはオフライン（期限切れ）の日付がありません。デフォルトでは、新しいバージョンがオンラインになるとプロセスバージョンは自動的にオフラインになります。
デプロイされた日付	プロセスがエンジンデータベースに追加された日付。
プロセス	アクティブなプロセスインスタンスの数。
移行先	このバージョンの移行先であるプロセスバージョン。
ステータス	オンライン、オンライン保留中、オフライン、オフライン保留中。

## [デプロイされたプロセスバージョンの詳細] ページ

【デプロイされたプロセスバージョンの詳細】 ページには、プロセスデプロイメント記述子のすべての詳細情報とプロセス正義が表示されます。

Version Detail
BPEL

Migrated To:  
Plan Id: 132847714504282112  
Deployed Date: 2018/01/02 07:39 PM  
Online Date: 2018/01/02 07:39 PM  
Offline Date:    
Running Process Disposition: Maintain  
Status: Online  
Logging Level:   
Persistence Type: Brief  
Exception Management Type: engine  
Invoke Recovery Type: engine  
Process Instance Retention:  Days  
Process Idle Timeout (seconds):

My Role

Partner Link	Type	Allowed Roles	Role	Binding	Service	Policy
ProcessConsumer	spiPartnerLinkType		Consumer	MSG	0CBIPXmwe8dKkH9w443SkPKAgentCloud	<a href="#">view</a>

Partner Role

Partner Link	Type	Role	Linkage
HostProvider	avHostEnvironmentRuntimeAccessPLT	HostRuntimeProvider	<a href="#">static</a>
ServiceProvider_bpi06jn	spiPartnerLinkType	Consumer	<a href="#">static</a>

Indexed Properties  
(None)

Resource Usage

Resource	Version	Target Namespace
AsInfoFunctions	6.0	urn:information.com:socrates:info:functions

このページのセクションについては、次のトピックで説明しています。

- プロセスバージョンのライフサイクル
- プロセスバージョンの更新
- プロセスバージョンの設定（オフラインまたはオンライン）
- ログレベル
- プロセスバージョンの永続タイプ
- 例外管理タイプ
- リカバリタイプの呼び出し
- プロセスインスタンスの保持期間
- デプロイされたプロセスの詳細グラフ

[デプロイされたプロセスバージョンの詳細] ページには、エンドポイント参照の詳細情報や他の関連する詳細情報も表示されます。

ロール	説明
マイロール	デプロイメント記述子の情報から、[マイロール] パートナーリンクエンドポイント参照の詳細情報が生成されます。パートナーリンクタイプにマウスを置くと、関連付けられた名前空間を表示できます。[サービス名] リンクを選択すると、パートナーリンクによって公開される Web サービスの WSDL ファイルを表示できます。[ポリシー] カラムで [表示] を選択すると、添付されているポリシーアサーションを表示できます。
パートナーロール	デプロイメント記述子の情報から、[パートナーロール] パートナーリンクエンドポイント参照の詳細情報が生成されます。パートナーリンクタイプにマウスを置くと、関連付けられた名前空間を表示できます。[リンケージ] カラムの静的エンドポイントタイプを選択すると、エンドポイント定義を表示できます。
インデックス付きプロパティ	インデックス付きプロパティが表示されます（ある場合）。

ロール	説明
イベントフィルタ	イベントフィルタが表示されます（ある場合）。イベントフィルタは、処理目的でイベントマネージャに渡されるプロセスイベントを指定します。カタログにデプロイされるイベントサービスは、イベントに対して動作します。
リソースの使用状況	「リソースの使用状況」には、WSDL、スキーマ、およびこのプロセスで参照されるその他のファイルとそのターゲット名前空間が表示されます。リソースを選択すると、リソース定義を表示できます。

## プロセスバージョンのライフサイクルについて

プロセスのバージョン管理により、プロセスサーバーにさまざまなバージョンのプロセスを置くことができます。プロセスのバージョン管理を使用すると、プロセスを有効にするタイミングや期間を制御できます。また、新しいバージョンが有効になったときに、古いバージョンで作成されたプロセスをどのように扱うかを制御することもできます。複数のプロセスバージョンが同時に存在することは可能ですが、新しいプロセスインスタンスを作成できるのは、最新の有効なバージョンのみです。

最新の有効なバージョンはオンラインの状態になっています。他には次のような状態があります。

- **オンライン保留:** 有効日が将来の日付になっているバージョンを示します。
- **オフライン保留:** 有効期限が切れているが実行中のプロセスインスタンスは引き続きアクティブになっているバージョンを示します。
- **オフライン:** 実行中のプロセスインスタンスがなくなった、期限切れのバージョンを示します。

プロセスデプロイメント記述子（PDD）には、デプロイメントがどのようにバージョン付けされるかを記述する属性を含むバージョン要素を含めることができます。これらの選択はすべてオプションで、次に示すようなデフォルト値があります。

次の例は、.pdd ファイルのバージョン情報の構文を示しています。

```
<version effectiveDate="2005-12-12T00:00:00-05:00"
  expirationDate="2007-12-12T00:00:00-05:00"
  id="1.5"
  runningProcessDisposition="migrate"/>
```

詳細は次のとおりです。

- **effectiveDate** は、新しいバージョンが最新バージョンになった日付で、新しいプロセスインスタンスはすべてこれに対して実行されます。  
実行中のプロセスに対して選択した処理によっては、一部は古いバージョンの使用が終わるまで引き続き実行されます。有効日は、XML スキーマの date/time 値です。日付表現にはタイムゾーンが含まれます。タイムゾーンは、コンピュータのタイムゾーンを午前 0 時と協定世界時（UTC）との時差で示した表記で表されます。上の例では、コンピュータのタイムゾーンは、UTC より 5 時間遅れた東部標準時です。有効日を指定しない場合は、プロセスがサーバーにデプロイされた日時にデフォルト設定されます。
- **expirationDate** は、現在のバージョンの期限が切れる、オンライン日より先の日付です。  
オフラインバージョンは、新しいプロセスインスタンスを作成できません。オフライン保留バージョンに結び付けられた実行中のすべてのプロセスが完了すると、バージョンはオフラインになります。現在のバージョンのすべてのプロセスインスタンスは、完了するまで実行されます。有効期限は、XML スキーマの date/time 値です（有効日と同じ）。有効期限を指定しない場合、アプリケーション統合コンソールでオフラインに手動設定するか、新しいバージョンがデプロイされるまで、バージョンは期限切れになりません。
- **id** は、major.minor 形式で表すプロセスバージョン番号です。

バージョン番号は、プロセスサーバーによって自動的に新しいバージョンに番号付けされるため、指定する必要はありません。サーバーは、minor 値を減らして major 値を 1 つ増やすことで、バージョン番号を増分します。例えば、バージョン 1.5 はバージョン 2.0 に増分されます。

- `runningProcessDisposition` は、このバージョンの有効日の到来後に、プロセスサーバーが、実行中のプロセスを持つ同じプロセスの他のバージョンに対して実行するアクションです。  
`runningProcessDisposition` の値は次のとおりです。
  - Maintain: 以前のバージョンのすべてのプロセスインスタンスは、完了するまで実行する必要があります。これがデフォルト値です。
  - Migrate: 以前のバージョンで作成された実行中のすべてのプロセスインスタンスは、状態情報を移行し、有効日が到来したときに新たにデプロイされたプロセス定義を使用します。バージョン間に互換性のない変更がある場合、説明を伴う警告メッセージがアプリケーション統合コンソールのサーバーログに書き込まれます。詳細については、警告が生成された場合に実行中のプロセスを移行する方法を参照してください。
  - Terminate: 以前のバージョンで実行中のすべてのプロセスインスタンスは、プロセスインスタンスが完了したかどうかに関係なく、新しいバージョンの有効日に終了する必要があることを示します。

## プロセスバージョンの更新

このプロパティは [\[デプロイされたプロセスバージョンの詳細\]](#) ページで表示できます。

プロセスバージョンを更新するためのオプションは次のとおりです。

- バージョンのステータス（オンライン、オンライン保留、オフライン、オフライン保留）に応じて、オンラインの日付、オフラインの日付、および実行中のプロセスの処理を更新できます。例えば、オンラインバージョンに [\[オフラインの日付\]](#) を追加して、[\[更新\]](#) を選択できます。
- [\[プロセスインスタンスの保持日数\]](#) を設定して、プロセスバージョンを更新することもできます。詳細については、「[プロセスインスタンスの保持期間](#)」を参照してください。

## プロセスバージョンの設定（オフラインまたはオンライン）

このプロパティは、[「\[デプロイされたプロセスバージョンの詳細\] ページ」 \(ページ 89\)](#) ページで表示できます。

プロセスバージョンをオフラインまたはオンラインに設定するオプションは次のとおりです。

- 指定した時刻にこのバージョンを非アクティブ化するには、[\[オフラインの日付\]](#) フィールドに値を入力します。
- このバージョンを即座に非アクティブ化するには、[\[オフラインに設定\]](#) を選択します。このバージョンから新しいプロセスインスタンスをインスタンス化することはできません。実行中のすべてのプロセスが完了するまで、ステータスは [\[オフライン保留\]](#) です。
- 以前にこのバージョンをオフラインにした場合は、これをリストアできます。このバージョンをオンラインまたはオンライン保留（将来）のバージョンにリストアするには、使用可能なオプションである [\[オンラインにリストア\]](#) または [\[オンライン保留にリストア\]](#) を選択します。

## ログレベルの設定

このプロパティは [\[デプロイされたプロセスバージョンの詳細\]](#) ページで設定できます。

実行中または完了したプロセスについては、実行ログを表示またはダウンロードできます。実行ログにはアクティビティの実行の開始時間と終了時間が記載されており、フォールトになったプロセスのトラブルシューティングに役立ちます。次のリストは、さまざまなログレベルを示しています。

- **なし:** プロセスサーバーは情報をログに記録しません。このオプションを使用すると、エンジンのパフォーマンスが向上します。
- **フォールト:** プロセスサーバーはフォールト情報のみログに記録します。プロセスでフォールトが発生しない場合、プロセスのログレベルはデフォルトで **【なし】** になります。**【フォールト】** を選択すると、ログファイルのサイズを縮小できます。
- **実行:** これはデフォルトのオプションです。プロセスサーバーは、*Will Not Execute* 文を除くすべての実行文をログに記録します。**【実行】** を選択すると、ログファイルのサイズを減らせます。
- **サービスデータを使用した実行:** プロセスサーバーは、すべての実行情報、フォールト情報、およびいくつかの WSIO アクティビティ情報をログに記録します。実行情報については、プロセスサーバーはデッドパスの状態、終了、実行準備完了などのログを記録します。WSIO の場合、プロセスサーバーは、呼び出し、選択、および受信をログに記録します。ただし、データの割り当てや変更に関する情報は記録しません。
- **データを使用した実行:** プロセスサーバーは、すべての実行文（*Will Not Execute* 文を除くが、変数、式、パートナーリンクデータは含む）をログに記録します。**【データを使用した実行】** を選択すると、ログファイルのサイズを減らせます。
- **完全:** プロセスサーバーは、デッドパスアクティビティのすべての実行文と *Will Not Execute* 文をログに記録します。例えば、プロセスサーバーは、実行されないフォールト処理文をすべてログに記録します。
- **システムのデフォルト:** このオプションを使用すると、プロセスバージョンのログレベルは、対応するプロセスサーバーで設定されているプロセスログレベルプロパティに対応します。

処理速度を高めるには、ログを実行しないか、最小限に抑えます。Informatica では、ログレベル **【なし】** または **【簡易】** の使用を推奨します。

## プロセスバージョンの永続タイプの設定

このプロパティは、[「\[デプロイされたプロセスバージョンの詳細\] ページ」 \(ページ 89\)](#) ページで設定できます。

永続性とは、アクティブなプロセスのストレージを指します。デフォルトでは、プロセスがサーバー上で実行されると、すべての状態と変数データがプロセスサーバーデータベースに保存されます。ただしこの設定は、サーバーパフォーマンスを向上し、データベースサイズを縮小するように PDD ファイルで変更できます。

永続性設定には次のような選択肢があります。

設定	説明
完全 (デフォルト)	各プロセスインスタンスについて、実行中、フォールト、完了のすべての状態の情報が保存されます。サーバーエラーが発生した場合、実行中のプロセスは完全にリカバリできます。リカバリが可能なのは、この設定によりプロセスサーバーがジャーナル（データベース向けの変更のレコード）を管理しているためです。  <b>注:</b> プロセスが、パートナーロールに WS-RM 呼び出しハンドラを使用している場合、またはマイロールに WS-Reliable メッセージングのポリシーアサーションを使用している場合、完全な永続性が求められます。
永続	<b>【完全】</b> と同じストレージ設定ですが、ジャーナルを作成しない点が異なります。プロセスが実行中でサーバーで障害が発生すると、プロセスは一時停止状態になります。  プロセスは、システムが停止してもリカバリ可能ですが、ジャーナルを作成していないため確認が必要です。このプロセスは一時停止中としてマークされます。

設定	説明
最終	プロセスの最終状態（完了またはフォールト）およびプロセス変数のみを保存します。サーバーエラーが発生すると、実行中のプロセスは終了します。この設定ではデータベースへの書き込みが前の2つの設定より少ないものの、アプリケーション統合コンソールの「アクティブなプロセスの詳細」ページでは、プロセスのグラフを表示できます。ここでは、実行パスとプロセス変数の最終値を確認できます。プロセスはメモリ内でのみ実行され、[プロセスアイドルタイムアウト]というサーバープロパティはこの永続レベルには何の影響もありません。
簡易	これは、プロセスログの最低レベルです。プロセスサーバーは、開始時刻と完了時刻、および最終状態（完了またはフォールト）のみを格納します。また、プロセスがフォールトになった場合のみ、状態とプロセス変数を格納します。プロセスはメモリ内でのみ実行され、[プロセスアイドルタイムアウト]というサーバープロパティはこの永続レベルには何の影響もありません。
なし	プロセスが終了しても、サーバーデータベースにはプロセス情報は一切格納されません。プロセスインスタンスは、[プロセス] ページにはリストされません。

## リカバリタイプの呼び出し

このプロパティは、「[\[デプロイされたプロセスバージョンの詳細\] ページ](#)」（[ページ 89](#)）ページで表示できます。

ノードエラーが原因で完了しない呼び出しアクティビティについては、リカバリ時にプロセスを一時停止できます。プロセスは保留中の呼び出しで一時停止されますが、プロセス例外管理を実行できます。

プロセスデプロイメント記述子（PDD）ファイルのエントリを使用して、エンジン設定を個別のプロセスで上書きできます。設定は以下のとおりです。

設定	説明
システムのデフォルト	すべてのプロセスの現在のエンジン設定。デフォルトのエンジン設定は呼び出しリカバリを無効にすることです。しかし、現在の設定は異なる場合があります。
False	保留中の呼び出しアクティビティはリカバリ時に一時停止できません。プロセスは異常終了します。この設定はエンジン設定を上書きします。
True	リカバリ時に保留中の呼び出しを一時停止し、一時停止中（フォールト）状態にします。その後、呼び出しアクティビティに対し、プロセス例外管理（フォールト状態のアクティビティやスコープの再試行や完了など）を実行できます。この設定はエンジン設定を上書きします。

## プロセスインスタンスの保持期間

[\[デプロイ済みプロセスバージョン詳細\]](#) ページで「**プロセスインスタンス保持時間**」プロパティを表示できます。

自動化されたスケジュールで削除するまで、完了したプロセスとフォールトになったプロセスをプロセスサーバーデータベースに保存する期間を指定できます。この設定は、次の場所で使用できます。

- Process Developer のプロセスデプロイメント記述子エディタ。プロセスのデプロイメント記述子に設定を追加できます。
- アプリケーション統合コンソールの「デプロイ済みプロセスバージョン詳細」ページ。デプロイされたプロセスについて、設定を追加または変更します。
- アプリケーション統合コンソールには、すべてのプロセスを対象としたデフォルトの保持日数設定。個々のプロセスに対する設定は、この設定より優先されます。

保持設定はステータスがオンラインまたはオンライン保留中のプロセスバージョンに適用され、オフライン保留またはオフラインのバージョンには適用されません。スケジュールは、**【プロセス】** ページに示されるように、各プロセスの完了日またはフォールト日に開始します。例えば、プロセスインスタンスが 12 月 31 日に完了し、保持設定が 30 日の場合、プロセスインスタンスは、次に削除がスケジュール設定されている 1 月 30 日以降にデータベースから削除されます。

#### 個々のプロセスの保持の設定

1. アプリケーション統合コンソールで、**【デプロイされたアセット】** > **【デプロイされたプロセス】** を選択します。
2. リストからプロセスを選択します。
3. 複数のバージョンが存在する場合は、**【デプロイ済みプロセス詳細】** ページでオンラインまたはオンライン保留中のバージョンを選択して、**【デプロイ済みプロセスバージョン詳細】** ページを開きます。
4. **【プロセスインスタンス保持時間】** フィールドで、そのプロセスバージョンに対する保持日数、時間数、分数を指定します。
5. **【更新】** を選択します。変更は、完了したプロセスと障害が発生したすべてのプロセスに対して直ちに有効になります。

スケジュール設定したデータベースメンテナンスの詳細については、ストレージの説明を参照してください。

## デプロイされたプロセスの詳細グラフ

**【デプロイされたプロセスの詳細】** グラフページは、プロセスインスタンスに関するさまざまな詳細情報を示します。

- **【アウトライン】** ビューには、BPEL プロセスの構造上の要素が表示されます。要素を選択して、そのプロパティと値を確認できます。
- **【グラフィック】** ビューには、メインのプロセスフローが表示されます。プロセスにイベントハンドラ、フォールトハンドラ、および補償ハンドラがある場合は、タブを選択することでそれらを表示できます。アクティビティを選択してそのプロパティを表示することもできます。
- **【プロパティ】** ビューは、選択した要素について表示されます。例えば、**【アウトライン】** ビューから **【プロセス】** 要素が選択されると、プロセスのプロパティと現在の値が表示されます。

## プロセスの再開

管理者またはオペレータのロールを持つユーザーは、完了したプロセス、フォールトになったプロセス、または一時停止したプロセスをアプリケーション統合コンソールから再開できます。アプリケーション統合の **【自分のプロセス】** タブには、認証で実行したすべてのプロセスのリストが表示されます。この **【自分のプロセス】** タブから、完了したプロセス、フォールトになったプロセス、または一時停止したプロセスを再開できます。

次の条件を満たすプロセスを再開できます。

- プロセスがクラウドサーバーまたはエージェントで少なくとも 1 回実行されている。
- プロセスが次のいずれかの状態である。
  - 完了
  - フォールト
  - 一時停止中
- プロセスのトレースレベルが、**【標準】** または **【詳細】** に設定されている。

- プロセスに添付ファイルが含まれている場合、持続性のタイプが【フル】または【永続】に設定されている。

【バインディング】プロパティが【イベント】に設定されているプロセスを再開することもできます。

複数のステップで【インスタンスの作成】プロパティが【はい】に設定されている BPEL プロセスを再開することはできません。

プロセスを編集しても実行しなかった場合、そのプロセスを再開すると、最後に実行したプロセスバージョンが再開されます。

サブプロセスを含むプロセスを再開すると、サブプロセスも再開されます。親プロセスに複数のサブプロセスが含まれている場合、特定のサブプロセスを再開するかどうかを選択できます。

## アプリケーション統合コンソールからのプロセスの再開

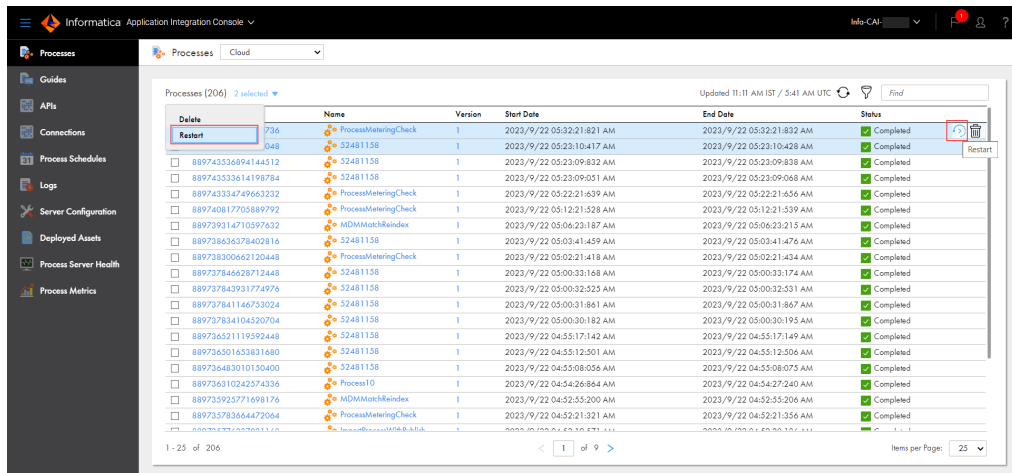
アプリケーション統合コンソールから、完了したプロセス、フォールトになったプロセス、または一時停止したプロセスを再開できます。

- アプリケーション統合コンソールで、左側のナビゲーションペインにある【プロセス】をクリックします。
- 再開するプロセスを選択し、プロセスインスタンスを含む行の【再開】アイコンをクリックします。

複数のプロセスを同時に再起動するには、プロセスを選択し、次のいずれかの手順を実行します。

- 【選択済み】リストで【再起動】を選択します。
- 選択した行を右クリックして、【再起動】を選択します。

次の図は、【プロセス】ページで利用可能な【再起動】オプションを示しています。



選択したプロセスが再開され、異なる実行 ID でリストされます。実行 ID をクリックすると、プロセス実行の詳細を表示できます。

## アプリケーション統合からのプロセスの再開

アプリケーション統合の【自分のプロセス】タブには、認証で実行したすべてのプロセスのリストが表示されます。この【自分のプロセス】タブから、完了したプロセス、フォールトになったプロセス、または一時停止したプロセスを再開できます。

次のビデオは、アプリケーション統合からプロセスを再開する方法を示しています。

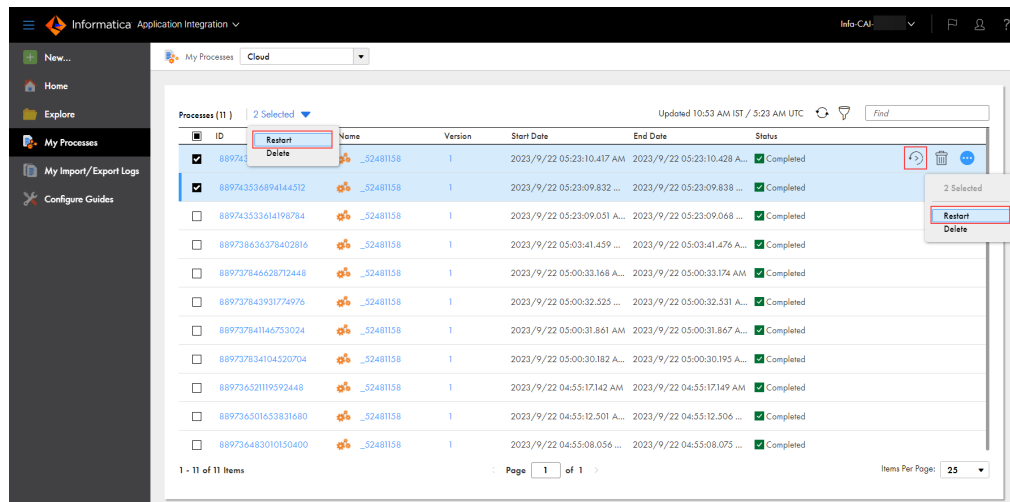
<https://knowledge.informatica.com/s/article/DOC-18507>

- アプリケーション統合で、左側のナビゲーションペインにある【自分のプロセス】をクリックします。

2. 再開するプロセスを選択し、プロセスインスタンスを含む行の【再開】アイコンをクリックします。  
複数のプロセスを同時に再起動するには、プロセスを選択し、次のいずれかの手順を実行します。

- 【アクション】メニューで【再起動】を選択します。
- 【選択済み】リストで【再起動】を選択します。
- 選択した行を右クリックして、【再起動】を選択します。

次の図は、【自分のプロセス】ページで利用可能な【再起動】オプションを示しています。



選択したプロセスが再開され、異なる実行 ID でリストされます。実行 ID をクリックすると、プロセス実行の詳細を表示できます。

## 第 6 章

# プロセスサーバーの正常性

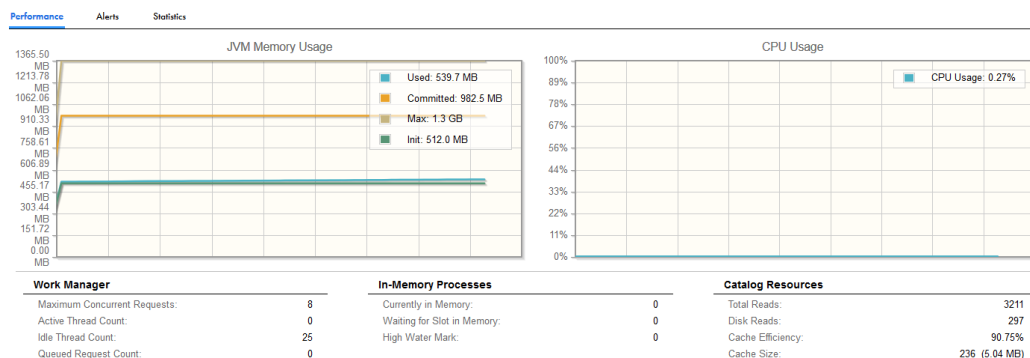
プロセスサーバーの正常性の詳細を表示するには Secure Agent を選択する必要があります。

## パフォーマンス

以下の詳細が［パフォーマンス］タブに表示されます。

- 全体的なパフォーマンスのデータ。
- サービス要求のパフォーマンスおよびスループット。
- リソースとサービスのボトルネック。

以下に例を示します。



ページを初めて開くと、メモリ内のスナップショットが統計に表示されます。最新の内容を表示するには、ページを更新します。

統計情報により、次のことがわかります。

- サーバーに送信している要求の数
  - キューの長さ
  - 待機中の要求の数
  - 応答にかかる時間
  - サーバーがサービス呼び出しから戻るまでにかかる時間
  - [サーバーのプロパティ] ページで設定された作業マネージャの制限値を増やす必要があるかどうか
- ロードテストの実行後、メトリックを監視し、設定プロパティを設定またはリセットできます。

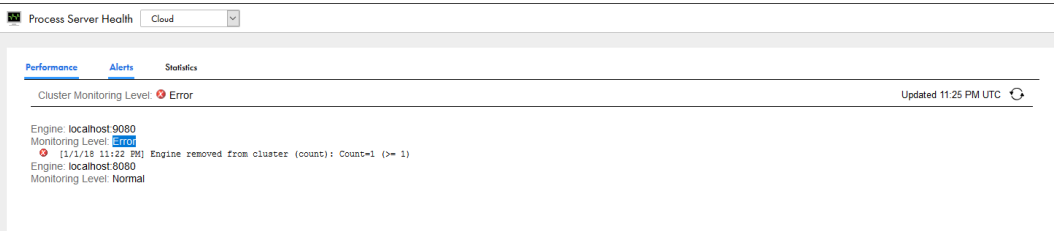
### プラットフォームのサポート

アプリケーションサーバーに関連してパフォーマンスを分析する際は、次の点に留意します。

- 他の設定を変更する前に、次の統計情報を考慮する必要があります。
  - CPU 使用率—時間間隔ごとのグラフ
  - JVM メモリ使用率—時間間隔ごとのグラフ
  - データベース接続プールの使用可能、使用中、待機中、高ウォーターマーク

# アラート

サーバー統計監視しきい値によってトリガされたアラートが表示されます。



しきい値間隔の期間に発生したエラーまたは警告は、評価期間の最後に報告されます。例えば、しきい値間隔が 5 分、評価頻度が各間隔で 5 回と設定されている場合、最初のトラブル項目は 1 分経過以降に現れますが、それが特定のしきい値間隔の期間に複数回報告されることはありません。同じ評価サイクルの期間に、競合するしきい値レベルが発生すると、最も高い重大度が報告されます。

監視に関する統計情報は、ブラウザを更新するたび、または「監視」ページを開くたびに更新されます。ただし、レポートが更新されるのは、評価期間の最後のみです。

トラブル項目のリストには、報告された各項目についてのエラー/警告レベル、時間、プロパティ名、および設定詳細が表示されます。

# 統計

**注:** このコマンドは、エージェントを選択した場合にのみ表示されます。

統計ページを使用すると、自分で設定した、またはデフォルトの監視プロパティを表示できます。プロセスサーバーは、エンジンの統計情報を収集し、それらを間隔別に集計します。必要に応じ、しきい値間隔や、各プロパティのエラー/警告レベルを設定できます。これらを設定すると、監視対象のプロパティであることを示すメガネアイコンなど、追加のシグナルがこのページに表示されます。

Server Statistics 1 selected									
Updated 3:23 PM IST									
Property Name	Monitored	Current Interval (5 min.)				Since Start or Reset			
		Samples	Min	Max	Avg	Samples	Min	Max	Avg
<input checked="" type="checkbox"/> Faulted/Faulting processes (count)		0	---	---	---	0	---	---	---
<input type="checkbox"/> Process cache efficiency (percent)		0	---	---	---	---	---	---	---
<input type="checkbox"/> Time to obtain process (ms)		0	---	---	---	---	---	---	---
<input type="checkbox"/> Work manager work start delay (ms)		0	---	---	---	---	---	---	---
<input type="checkbox"/> Database connection acquisition time (ms)		7	0	1	0	7	0	1	0
<input type="checkbox"/> Discarded unmatched correlated receives (count)		0	---	---	---	0	---	---	---
<input type="checkbox"/> Time to validate messages (ms)		0	---	---	---	---	---	---	---
<input type="checkbox"/> Critical storage exceptions (count)		0	---	---	---	0	---	---	---
<input type="checkbox"/> Deadlock retry attempts (count)		0	---	---	---	0	---	---	---

Individual Engine Monitors

Engine: INW00007327									
Property Name	Monitored	Current Interval				Since Start or Reset			
		Samples	Min	Max	Avg	Samples	Min	Max	Avg
Faulted/Faulting processes (count)		0	---	---	---	0	---	---	---
Process cache efficiency (percent)		0	---	---	---	---	---	---	---
Time to obtain process (ms)		0	---	---	---	---	---	---	---
Work manager work start delay (ms)		0	---	---	---	---	---	---	---
Database connection acquisition time (ms)		7	0	1	0	7	0	1	0
Discarded unmatched correlated receives (count)		0	---	---	---	0	---	---	---

収集間隔の期間、プロセスサーバーは、最大値、平均値、合計数など、設定した各プロパティ値の統計を管理します。さらに、履歴統計も収集します。履歴統計には、記録中の間隔での最小値や最大値が含まれます。統計情報はメモリ内に収集されます。

統計をクリアするには、**【リセット】**を選択します。例えば、エラー状態を修正した場合は、統計の収集をリセットしてゼロ値に戻し、パフォーマンスが向上したかどうかを確認できます。エラーと警告をすべて解決すると、エンジンの監視レベルは **【ノーマル】** にリセットされます。

## 第 7 章

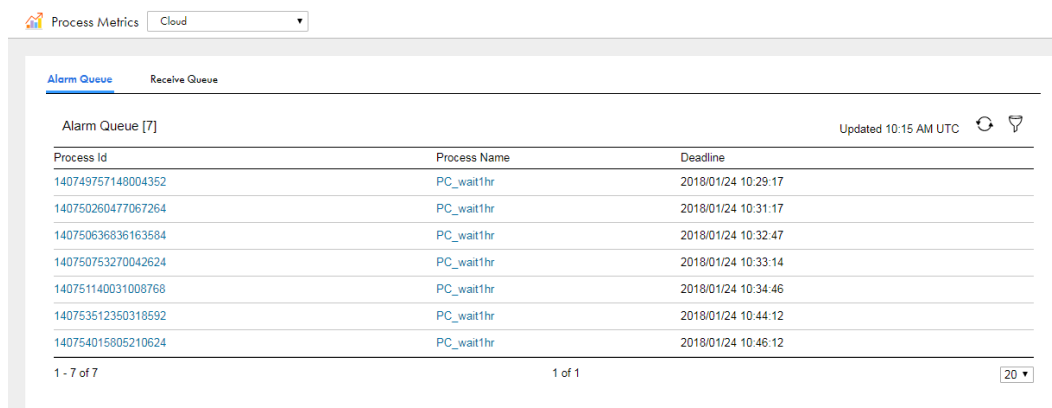
# プロセスメトリック

アラームキューおよび受信キューのメトリックをフィルタリングおよび表示します。

## アラームキューメトリック

**[Alarm Queue (アラームキュー)]** ページには、アラートキュー内のアラームの一覧がプロセスごとに表示されます。

次の図に、**[Alarm Queue (アラームキュー)]** ページの例を示します。



Process Id	Process Name	Deadline
140749757148004352	PC_wait1hr	2018/01/24 10:29:17
140750260477067264	PC_wait1hr	2018/01/24 10:31:17
140750636836163584	PC_wait1hr	2018/01/24 10:32:47
140750753270042624	PC_wait1hr	2018/01/24 10:33:14
140751140031008768	PC_wait1hr	2018/01/24 10:34:46
140753512350318592	PC_wait1hr	2018/01/24 10:44:12
140754015805210624	PC_wait1hr	2018/01/24 10:46:12

選択フィルタを使用して、発生中のアラームを閲覧するときに表示するプロセスアクティビティを決定できます。

オプション	説明
有効期限	アラームの開始日時と終了日時。
プロセス ID	プロセスインスタンス ID。この ID は、[プロセス] ページで確認できます。
プロセス名	プロセス修飾名 (qname) のローカル部分。

オプション	説明
グループ	このプロセスが属するグループ（オプション）。グループ名は [On Alarm (オンアラーム)] リストに表示されます。
システムを非表示	デフォルトのシステムプロセスを表示する場合はチェックをオフにします。システムプロセスはデフォルトで非表示になっています。

## 受信キューメトリック

Alarm Queue <u>Receive Queue</u>			
Receive Queue [3] <span>🔄</span> <span>🔍</span>			
Process Id	Partner Link	Port Type	Operation
138242539990814720	ReceiveEventPL	spiPortType	Initiate
139966694478057472	ReceiveEventPL	spiPortType	Initiate
139967668898762752	ReceiveEventPL	spiPortType	Initiate
1 - 3 of 3		1 of 1	20 ▾

フィルタのオプションリストから 1 つ以上のオプションを選択し、選択したアクティブな受信を表示します。次をクリックします。

- **プロセス ID:** [プロセスビューの詳細] ページが表示されます。
- **パートナーリンク:** [メッセージレシーバの詳細] ウィンドウが表示されます。

### Message Receiver Details

🔍 🔍 ✕

**PartnerLink Type:** ReceiveEventPL

**Port Type:** http://services.informatica.com/spi/2013/12/spi-process/spiPortType

**Operation:** Initiate

**Process ID:** 138242539990814720

**Correlation Data**

```

{http://services.informatica.com/spi/2013/12/spi-process/}correlationId=Receive_Input=qw
{http://www.activebpel.org/bpel/extension}securityContext=71KG1mxF13b131z4e20IKr
    
```

? Close

【メッセージレシーバの詳細】 ウィンドウには次の情報が表示されます。

項目	意味
プロセス ID	プロセスインスタンス ID。この ID は、[アクティブなプロセス] ページで確認できます。
パートナーリンク	受信キューにある項目のパートナーリンク。パートナーリンクは、2 つのパートナー間でのやりとりです。最も基本的な形式では、プロセスは外部サービスのパートナーリンクで、そこから要求を受信します。パートナーリンクは、特定のやりとりでの、プロセスが担うロール（ある場合）と、パートナーサービスが担うロール（ある場合）を定義します。
ポートタイプ	ポートタイプ。WSDL のポートタイプは、受信、返信、呼び出し（操作を含む）など、関連する操作のセットです。
操作	今実行した操作。

## 第 8 章

# API

**[API]** ページには、REST/SOAP バインディングを使用するパブリッシュされたプロセスの API のリストが、REST および SOAP エンドポイントとそのステータスとともに表示されます。**[API]** ページを使用して、パブリッシュされたプロセスの API をアクティブ化または非アクティブ化します。

API、つまり、REST エンドポイントと SOAP エンドポイントを **[API]** ページからアクティブ化または非アクティブ化できます。このオプションを使用すると、API を非アクティブ化およびアクティブ化するためにプロセスをパブリッシュ解除してから再パブリッシュする必要がなくなります。アプリケーションの統合から初めてプロセスをパブリッシュすると、API がアクティブ化され、アプリケーション統合コンソールの **[API]** ページに表示されます。後ほど **[API]** ページから API をアクティブ化または非アクティブ化すると、アプリケーションの統合によって、パブリッシュされたプロセスの **【プロパティの詳細】** ダイアログボックスの API ステータスも更新されます。API ステータスが非アクティブであるプロセスを再パブリッシュしても、API はアクティブになりません。**[API]** ページからアクティブ化する必要があります。アプリケーションの統合からプロセスをパブリッシュ解除にすると、API は **[API]** ページから削除されます。

**注:** アクティブ化または非アクティブ化は、スケジュールされたプロセスおよびサブプロセスには適用されません。スケジュールされたプロセス API のアクティブ化または非アクティブ化は、スケジュールには影響しません。プロセスにサブプロセスが含まれている場合は、親プロセスの API を非アクティブ化しても、その内部のサブプロセスは引き続き実行されます。

### 例

アプリケーション統合を使用して開発された API を介して Web ポータルで資金移動を提供している銀行について考えてみましょう。この銀行は、Oracle データベースを使用しており、将来の監査と監視のために取引情報を保存しています。

銀行は、データベースのメンテナンス中、受信要求に関する潜在的な問題を回避するためにトランザクションを一時的に停止したいと考えています。

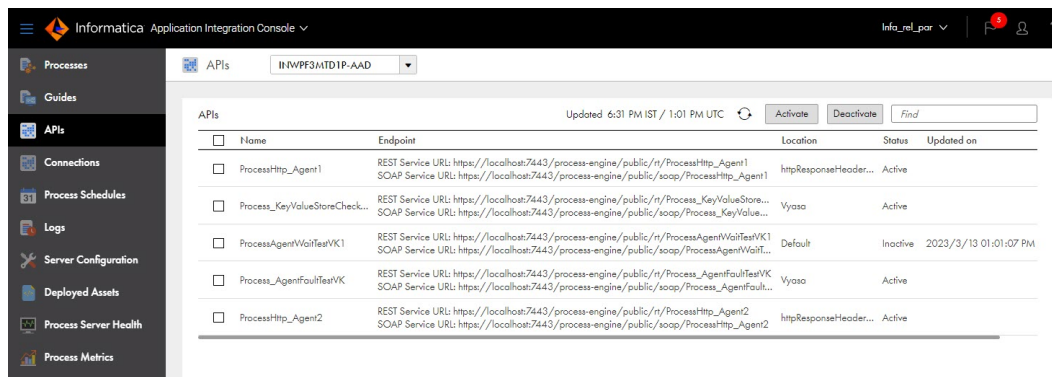
この場合、銀行は資金移動を可能にするすべての API をフィルタリングして見つけ、それらを無効にすることができます。ダウンタイム中に、顧客が非アクティブ化された API を呼び出すと、503 Service Unavailable または同等のメッセージが表示されます。メンテナンスが完了した後に、資金移動を再開するために API を有効にすることができます。

## API の表示

パブリッシュされたプロセスの API を表示するには、アプリケーション統合コンソールで **[API]** をクリックします。デフォルトでは、**[API]** ページにはクラウドサーバーでパブリッシュされたプロセスの API が一覧で表示されます。

クラウドサーバーまたは Secure Agent を選択し、プロセスの API 名を入力して API を検索できます。

次の図に、**[API]** ページを示します。



【API】 ページには、次の情報が表示されます。

#### 名前

クラウドサーバーまたは選択した Secure Agent でパブリッシュされたプロセスの名前が表示されます。

#### エンドポイント

REST および SOAP エンドポイント URL が表示されます。

#### 場所

接続が保存されているプロジェクトまたはフォルダの場所が表示されます。

#### ステータス

API のステータスが表示されます。ステータスは、次のいずれかの値になります。

- アクティブ。API がアクティブ化され、使用できる状態です。
- 非アクティブ。API が非アクティブ化されています。ユーザーが API を呼び出すと、503 Service Unavailable エラーが表示されます。

#### 更新日時

API ステータスが最後に更新された日時が表示されます。

## API の非アクティブ化

ユーザーがダイレクトエンドポイントを使用して API を実行しており、一定期間 API の使用を停止する場合は、API を非アクティブ化できます。API、つまり、パブリッシュされたプロセスの REST エンドポイントと SOAP エンドポイントを非アクティブ化できます。ユーザーが、非アクティブ化された API を呼び出すと、503 Service Unavailable エラーが表示されます。

API を非アクティブ化するには、次の手順を実行します。

1. アプリケーション統合コンソールで、左側のナビゲーションバーにある **【API】** をクリックします。
2. API を表示するには、クラウドサーバーまたは Secure Agent を選択します。
3. 非アクティブ化する 1 つ以上の API を選択し、**【非アクティブ化】** をクリックします。  
**【API ステータスの更新】** ダイアログボックスが表示されます。
4. クラウドサーバーで API を非アクティブ化するには、**【非アクティブ化】** をクリックして続行します。  
**【API】** ページで API のステータスが **【非アクティブ】** に更新されます。

5. Secure Agent で API を非アクティブ化するには、次のいずれかの手順を実行します。

- プロセスが、選択した Secure Agent でのみ実行されている場合は、**【非アクティブ化】** をクリックして続行します。
- グループ内の複数の Secure Agent でプロセスが実行されている場合は、次のいずれかの手順を実行します。
  - **【選択した Secure Agent でのみ非アクティブ化】** をクリックし、選択した Secure Agent を使用して続行します。
  - **【グループ内のすべての Secure Agent で非アクティブ化】** をクリックし、Secure Agent グループを使用して続行します。

**【API】** ページで API のステータスが **【非アクティブ】** に更新されます。

アプリケーション統合では、次の図に示すように、パブリッシュされたプロセスの **【プロパティの詳細】** ダイアログボックスの API ステータスも更新されます。

The image shows a dialog box titled "Properties Detail for HelloWorldProcess". It has a close button (X) in the top right corner. The dialog is divided into two main sections: "Basic" and "Endpoints".

**Basic**

Unique Name:	HelloWorldProcess
Location:	Default
Publication Status:	<input checked="" type="checkbox"/> Published
Published On:	2023-03-21 13:57
Published By:	
Applies To:	* Any *
Status:	Inactive

**Endpoints**

Service URL:	https://pod1-cai.rel.infaqa.com/active-bpel/public/rt/6CsVq4DWVvJf96ngDYdLSa/HelloWorldProcess <a href="#">"View Swagger File"</a> <a href="#">"View OpenAPI 3.0 File"</a>	<input type="button" value="Copy"/>
SOAP Service URL:	https://pod1-cai.rel.infaqa.com/active-bpel/public/soap/6CsVq4DWVvJf96ngDYdLSa/HelloWorldProcess <a href="#">"View WSDL File"</a>	<input type="button" value="Copy"/>

At the bottom left, there is a help icon (?). At the bottom right, there is a "Close" button.

**【プロパティの詳細】** ダイアログボックスで API のステータスが **【非アクティブ】** と表示されている場合は、API をアクティブ化して、ユーザーが API を使用できるようにする必要があります。

# API の有効化

非アクティブ化された API の使用を再開するには、API、つまり、パブリッシュされたプロセスの REST および SOAP エンドポイントをアクティブ化します。

API をアクティブにするには、次の手順を実行します。

1. アプリケーション統合コンソールで、左側のナビゲーションバーにある **[API]** をクリックします。
2. API を表示するには、クラウドサーバーまたは Secure Agent を選択します。
3. アクティブ化する 1 つ以上の API を選択し、**[アクティブ化]** をクリックします。  
**[API ステータスの更新]** ダイアログボックスが表示されます。
4. クラウドサーバーで API をアクティブ化するには、**[アクティブ化]** をクリックして続行します。  
**[API]** ページで API のステータスが **[アクティブ]** に更新されます。
5. Secure Agent で API をアクティブ化するには、次のいずれかの手順を実行します。
  - プロセスが、選択した Secure Agent でのみ実行されている場合は、**[アクティブ化]** をクリックして続行します。
  - グループ内の複数の Secure Agent でプロセスが実行されている場合は、次のいずれかの手順を実行します。
    - **[選択した Secure Agent でのみアクティブ化]** をクリックし、選択した Secure Agent を使用して続行します。
    - **[グループ内のすべての Secure Agent でアクティブ化]** をクリックし、Secure Agent グループを使用して続行します。

**[API]** ページで API のステータスが **[アクティブ]** に更新されます。

アプリケーション統合では、次の図に示すように、パブリッシュされたプロセスの **[プロパティの詳細]** ダイアログボックスの API ステータスも更新されます。

Properties Detail for HelloWorldProcess

Basic

Unique Name:

HelloWorldProcess

Location:

Default

Publication Status:

☒ Published

Published On:

2023-03-24 13:17

Published By:

Applies To:

\* Any \*

Status:

Active

Endpoints

Only accepts HTTP authorization requests from the API Gateway:

Service URL:

https://pod1-cai.rel.infaqa.com/active-bpel/rt/HelloWorldProcess

Copy

[View Swagger File](#)

[View OpenAPI 3.0 File](#)

SOAP Service URL:

https://pod1-cai.rel.infaqa.com/active-bpel/soap/HelloWorldProcess

Copy

[View WSDL File](#)

?

Close

## 第 9 章

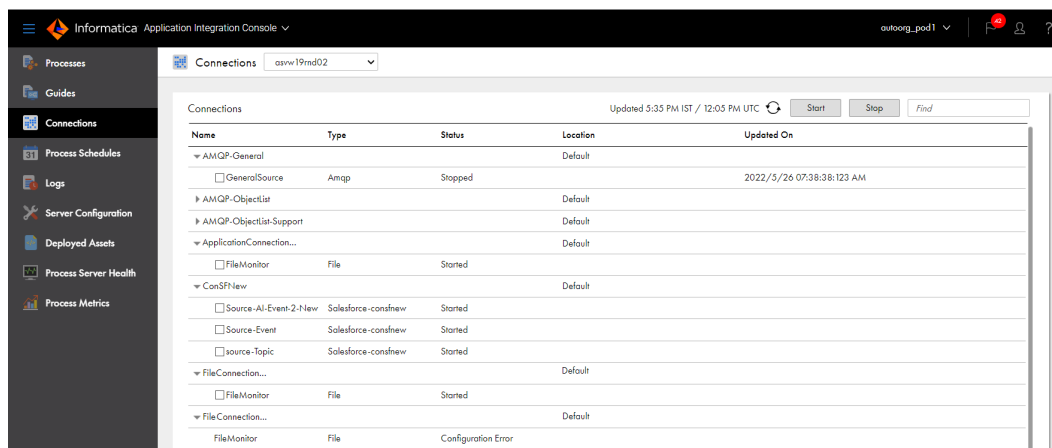
# 接続

**【接続】** ページには、選択した Secure Agent でパブリッシュされたリスナベース接続のイベントソースのリストが、接続タイプとステータスとともに表示されます。クラウドサーバーでパブリッシュされた Kafka 接続内のイベントソースのリストを表示することもできます。**【接続】** ページを使用して、リスナベース接続のイベントソースを表示、開始、または停止します。

**注:** Kafka 接続は、クラウドサーバー上の AWS POD でのみパブリッシュできます。

接続名または接続 ID を入力して、接続を検索できます。

次の図は、**【接続】** ページを示しています。



**【接続】** ページには、次の情報が表示されます。

### 名前

クラウドサーバーまたは選択した Secure Agent 上でパブリッシュされたリスナベースの接続を表示します。接続を展開すると、使用可能なイベントソースが表示されます。

### タイプ

イベントソースのタイプを表示します。

### ステータス

イベントソースのステータスを表示します。ステータスは、次のいずれかの値になります。

- 開始。イベントソースが開始しています。
- 停止。イベントソースが停止しています。
- 設定エラー。接続が誤った値でパブリッシュされました。例えば、IP アドレスに英数字が含まれている場合、アプリケーション統合はエンドポイントに接続できません。この場合は、設定値を修正して接続を再パブリッシュする必要があります。

## 場所

接続が保存されているプロジェクトまたはフォルダの場所を表示します。

## 更新日時

イベントソースが最後に更新された日時を表示します。

**【ログのダウンロード】** アイコンをクリックすると、クラウドサーバー上でパブリッシュされた Kafka 接続のリスナログをダウンロードできます。ログには、過去 24 時間の警告とエラーが含まれます。

# リスナベース接続でのイベントソースの開始と停止

Secure Agent または Secure Agent グループで実行されるリスナベース接続のイベントソースを開始および停止できます。クラウドサーバーでパブリッシュされた Kafka 接続内のイベントソースを開始および停止することもできます。このオプションを使用すると、イベントベースのプロセスを停止および開始するために、接続をパブリッシュ解除して再パブリッシュする必要はありません。

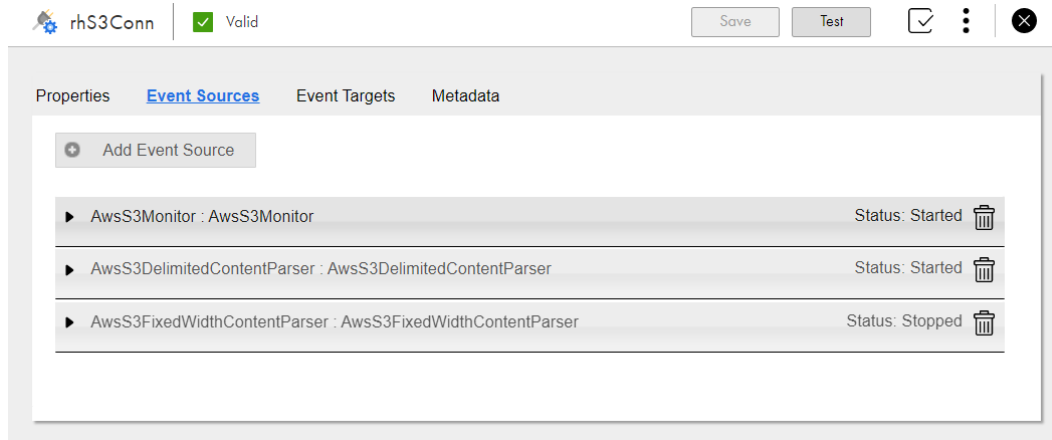
接続のイベントソースを開始および停止するには、次の手順を実行します。

1. アプリケーション統合コンソールで、左側のナビゲーションバーにある **【接続】** をクリックします。
2. **【クラウド】** または **Secure Agent** を選択して接続を表示します。
3. 接続を展開して、イベントソースを表示します。
4. 開始または停止するイベントソースを 1 つ以上選択し、**【開始】** または **【停止】** をクリックします。  
**【リスナステータスの更新】** ダイアログボックスが表示されます。
5. クラウドサーバー上でパブリッシュされている接続のイベントソースを開始するには、**【開始】** をクリックして続行します。  
**【接続】** ページで、イベントソースのステータスが **【開始】** に更新されます。
6. クラウドサーバー上でパブリッシュされている接続のイベントソースを停止するには、**【停止】** をクリックして続行します。  
**【接続】** ページで、イベントソースのステータスが **【停止】** に更新されます。
7. Secure Agent 上でパブリッシュされている接続のイベントソースを開始するには、次のいずれかの手順を実行します。
  - 選択した Secure Agent でのみ接続が実行される場合は、**【選択した Secure Agent で開始】** をクリックして続行します。
  - グループ内の複数の Secure Agent で接続が実行される場合は、次のいずれかの手順を実行します。
    - **【選択した Secure Agent でのみ開始】** をクリックし、選択した Secure Agent を使用して続行します。
    - **【グループ内のすべての Secure Agent で開始】** をクリックし、Secure Agent グループを使用して続行します。**【接続】** ページで、イベントソースのステータスが **【開始】** に更新されます。
8. Secure Agent 上でパブリッシュされている接続のイベントソースを停止するには、次のいずれかの手順を実行します。
  - 選択した Secure Agent でのみ接続が実行される場合は、**【選択した Secure Agent で停止】** をクリックして続行します。

- グループ内の複数の Secure Agent で接続が実行される場合は、次のいずれかの手順を実行します。
  - **【選択した Secure Agent でのみ停止】** をクリックし、選択した Secure Agent を使用して続行します。
  - **【グループ内のすべての Secure Agent で停止】** をクリックし、Secure Agent グループを使用して続行します。

**【接続】** ページで、イベントソースのステータスが **【停止】** に更新されます。

アプリケーション統合により、次の図に示すように、接続の **【イベントソース】** タブにあるイベントソースのステータスも更新されます。



クラウドサーバーで実行される接続では、イベントソースのステータスが **【停止】** の場合、接続を再パブリッシュしてイベントソースを再開できます。ただし、接続を再パブリッシュすると、接続内のすべてのイベントソースがデフォルトで開始されます。Secure Agent の場合、Secure Agent が稼動中であるときに限り、接続のすべてのイベントソースが開始されます。ステータスは、接続の **【イベントソース】** タブ、およびアプリケーション統合コンソールの **【接続】** ページで更新されます。接続をパブリッシュ解除すると、接続のすべてのアクティブなイベントソースは停止され、アプリケーション統合コンソールの **【接続】** ページから接続が削除されます。

**注:** 更新されたステータスを表示するには、接続の更新が必要になる場合があります。

リスナベース接続でのイベントソースの開始と停止の詳細については、「*Cloud アプリケーション統合用コネクタ*」を参照してください。

## 第 10 章

# ガイド

**【ガイド】** ページには、実行されたガイドのリストが表示されます。リストには、正常に完了したガイド、フォールトで終了したガイド、現在実行中のガイドが表示されます。

ガイドはこのページに次の期間表示されます。

- 正常に完了したガイド: 7 日間。
- フォールトで終了または早期に終了したガイド: 18 日間。
- 実行中のガイド: ガイドの実行中のインスタンスがある限り。すべてのインスタンスが完了すると、正常に完了したインスタンスまたはフォールトになったインスタンスがあるかどうかによって変わります。

実行されていないガイドは非アクティブとみなされます。非アクティブなガイドは 7 日後に自動的に終了します。非アクティブなガイドとマイルストーンステップにあるガイドは、14 日後に自動的に終了します。

**【ガイド】** ページには次の情報が表示されます。

名前	説明
名前	<b>【ガイドデザイナー】</b> > <b>【開始】</b> > <b>【全般】</b> に入力した名前。
現在のユーザー	現在ガイドを実行している組織のユーザーの数。 例えば、13 という値は実行中のガイドのインスタンスが 13 個あることを示します。
合計実行数	ユーザーがガイドを実行した回数。 <b>【合計実行数】</b> には、正常に完了した実行、フォールトになった実行、継続中の実行が含まれます。
完了した実行	ガイドが正常に完了した、またはフォールトで終了した回数。
停止	<b>【現在のユーザー】</b> の値がゼロでないガイドにマウスカーソルを重ねると、 <b>【停止】</b> オプションが表示されます。すべてのインスタンスを停止するには、 <b>【停止】</b> をクリックします。

次の図は、**【ガイド】** ページを示しています。

Guides

Guides (17) Updated 11:26 AM UTC Find

Add Field ▼

Name	Current Users	Total Runs	Completed Runs
AlterMeeting	13	29	16
Alter_Meeting	9	9	0
appliedToof-oomok	1	1	0
CreateUserCase	1	5	4
Current_User2	2	2	0
Guide1-1	1	1	0
Main1	1	7	3
Main2-1	3	6	0
Main3	2	2	0
Main3-1	2	2	0
Provision Organization	2	3	1
SF_Account	2	6	4
testCreateGuide	2	3	1
testGuideForCRUD	1	4	3
TestSimpleGuide	0	1	1
ThrowTest	0	2	0
translate_ukr	3	5	2

ガイド名をクリックすると、**【ガイド結果】** ページが表示されます。

## [ガイド結果] ページ

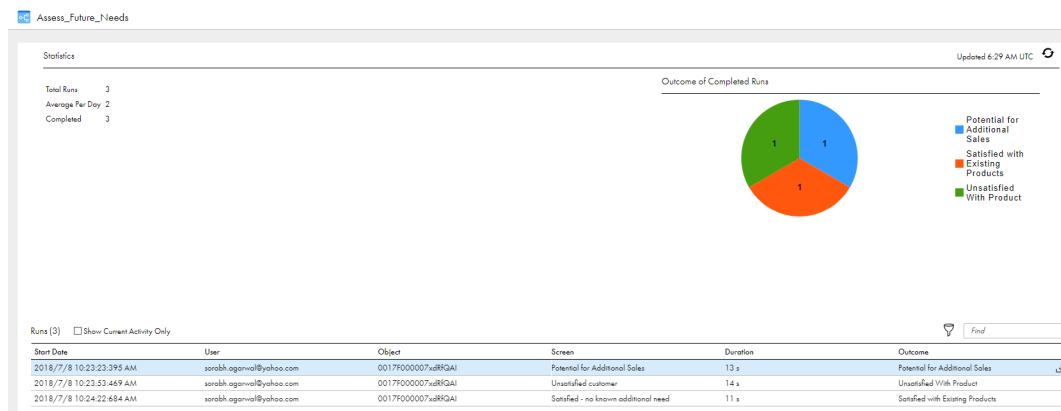
**【ガイド結果】** ページには、ガイドインスタンスに関する詳細が表示されます。

各ガイドインスタンスの次のプロパティを確認できます。

名前	説明
開始日	ユーザーがガイドを開始した日付。
ユーザー	<p><b>【ガイドデザイナー】 &gt; 【開始】 &gt; 【開始】 &gt; 【適用先】</b> で定義した内容によって異なります。ガイドの適用先を特定のオブジェクトではなく <b>【すべて】</b> のオブジェクトに定義した場合、<b>【ユーザー】</b> フィールドには、ガイドを実行した Informatica Intelligent Cloud Services ユーザーのログイン ID が含まれます。</p> <p>ガイドの適用先を特定の Salesforce オブジェクトに定義した場合、<b>【ユーザー】</b> フィールドには Salesforce ログイン ID が含まれます。</p>
オブジェクト	<p><b>【ガイドデザイナー】 &gt; 【開始】 &gt; 【開始】 &gt; 【適用先】</b> で定義したオブジェクトを含む Salesforce 組織の ID。</p> <p>ガイドの適用先を <b>【すべて】</b> のオブジェクトに定義した場合、<b>【オブジェクト】</b> セルにオブジェクト ID は表示されません。</p>
画面	<p>ガイドが停止した画面。</p> <p>ガイドが正常に完了すると、最後の画面の名前が表示されます。</p> <p>ガイドがフォールトになると、ガイドが停止した画面が表示されます。</p> <p>ガイドが実行中の場合、ユーザーが現在いる画面が表示されます。</p>
継続時間	<p>ガイドが正常に完了するかフォールトで終了するのにかった時間。</p> <p>ガイドが実行中の場合、<b>【継続時間】</b> セルに時間は表示されません。</p>

名前	説明
結果	<p>次の値のいずれかが表示されます。</p> <ul style="list-style-type: none"> <li>- <b>【ガイドデザイナー】 &gt; 【開始】 &gt; 【結果】</b> で定義した結果。ガイドインスタンスが正常に完了した場合に結果の値が表示されます。</li> <li>- <b>早期に終了しました。</b> ガイドが完了する前にユーザーによって停止された場合にこの値が表示されます。</li> <li>- <b>不明。</b> ガイドがフォールトになった場合にこの値が表示されます。</li> </ul> <p>ガイドが実行中の場合、<b>【結果】</b> には値は表示されません。</p>
アクション	ログをダウンロードしたり、実行中のガイドインスタンスを停止することができます。

次の図は、**【ガイド結果】** ページを示しています。



この図では、ユーザーは Asses\_Future\_Needs ガイドを 3 回実行し、それぞれ別の結果になっています。すべてのインスタンスが正常に完了しました。

# 第 11 章

## ログ

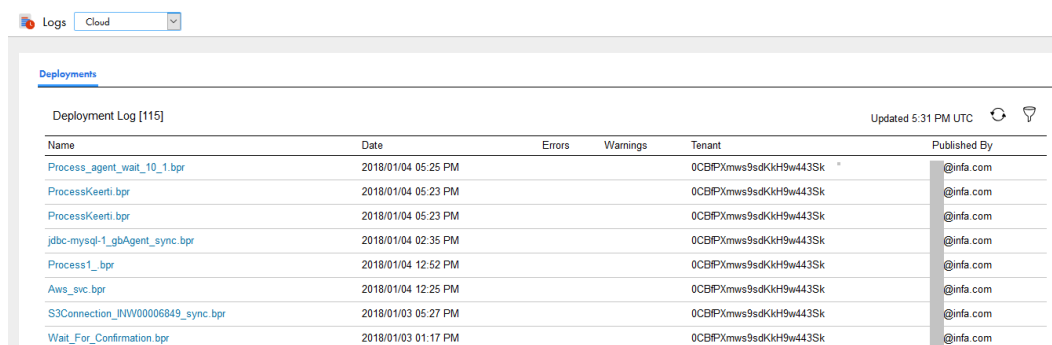
「ログ」ページには、サーバーログとデプロイメントログが表示されます。

サーバーログを表示するには Secure Agent を選択する必要があります。

## デプロイメントログ

「デプロイメントログ」タブには、新規または変更されたビジネスプロセスアーカイブ（.bpr）ファイルがデプロイされるときに生成されるログのリストが表示されます。

次の図は、「デプロイメントログ」タブのサンプルを示しています。



Name	Date	Errors	Warnings	Tenant	Published By
<a href="#">Process_agent_wait_10_1.bpr</a>	2018/01/04 05:25 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com
<a href="#">ProcessKeerti.bpr</a>	2018/01/04 05:23 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com
<a href="#">ProcessKeerti.bpr</a>	2018/01/04 05:23 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com
<a href="#">jdbc-mysql-1_gbAgent_sync.bpr</a>	2018/01/04 02:35 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com
<a href="#">Process1_.bpr</a>	2018/01/04 12:52 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com
<a href="#">Aws_svc.bpr</a>	2018/01/04 12:25 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com
<a href="#">S3Connection_INW00006849_sync.bpr</a>	2018/01/03 05:27 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com
<a href="#">Wait_For_Confirmation.bpr</a>	2018/01/03 01:17 PM			0CBFPXmws9sdKkH9w443Sk	@infa.com

生成されたエラーと警告（ある場合）の数が表示されます。

このページでは、次を実行できます。

- 選択フィルタを使用して、ログリストの表示を変更する
- .bpr ファイルを選択してデプロイメントログを表示する

.bpr ファイルをクリックすると、Process Developer によって「デプロイメントログ」ページが表示されます。このページには、選択した .bpr ファイルの名前、日付、およびログが表示されます。

## Deployment Log



```
Contribution:      project:/OCBFPXmws9sdKkH9w443Sk/spi.ProcessWait-1
Name:              ProcessWait_1.bpr
Date:              2016/01/03 11:47 AM
Log
[ProcessWait_1.bpr] [ProcessWait-1.pdd] ActiveVOS specific extension function used: getFaultString.
[ProcessWait_1.bpr] [ProcessWait-1.pdd] ActiveVOS specific extension function used: getProcessId.
[ProcessWait_1.bpr] [ProcessWait-1.pdd] ActiveVOS specific extension function used: copyAllAttachments.
[ProcessWait_1.bpr] [ProcessWait-1.pdd] ActiveVOS specific extension function used: getProcessInitiator.
[ProcessWait_1.bpr] [ProcessWait-1.pdd] ActiveVOS specific extension function used: setProcessTitle.
[ProcessWait_1.bpr] [ProcessWait-1.pdd] The declared expression language urn:active-endpoints:expression-language:xcquery1.0 is non-standard and may not be supported.
[ProcessWait_1.bpr] [ProcessWait-1.pdd] BPEL 2.0 extensions are used in this process.
[ProcessWait_1.bpr] [ProcessWait-1.pdd] Passed validation and was stored in database.
[ProcessWait_1.bpr] [ProcessWait-1.pd.xml] Added resource mapped to location hint: project:/spi.ProcessWait-1/ProcessWait-1.pd.xml
[ProcessWait_1.bpr] [entry-points.xml] Added resource mapped to location hint: project:/spi.ProcessWait-1/entry-points.xml
[ProcessWait_1.bpr] [services.xml] Added resource mapped to location hint: project:/spi.ProcessWait-1/services.xml
```

デプロイメントの間、エンジンは BPEL プロセスのデプロイメント記述子を検証し、その WSDL ファイルと他のリソースが使用可能で、現在のバージョンのプロセスに対して有効であることを確認します。検証時にエラーや警告が発生した場合は、修正を加えて .bpr ファイルを再デプロイするか、無効なプロセスに対して新しい .bpr ファイルを作成します。

**注:** コントリビューションを削除すると、デプロイメントログも削除されます。

### 選択フィルタ

1. 必要に応じ、ログコンテンツのタイプを選択します。
2. 必要に応じ、デプロイの期間の日付を選択します。
3. 必要に応じ、.bpr ファイルの名前を入力します。名前にはワイルドカード (\*) を使用できます。
4. **【送信】** を選択します。選択フィルタに基づいて、デプロイメントログのリストが再表示されます。

## サーバーログ

サーバーログはサーバーで発生するイベントを記録します。これにより、問題を確認してトラブルシューティングできます。各イベントについて、発生時刻、問題のレベル、ソース（イニシエータ）、サービス（イベントタイプ）、エンジン、およびイベントを説明するメッセージがログに記録されます。例えば、BPR デプロイメントでは、情報（レベル）、システム（ソース）、エンジン（サービス）のイベントが記録されます。

**注:** サーバーログは、イベントが発生しても即座に更新されない場合があります。通常は、イベント発生後、30 秒から 60 秒の間に更新されます。あるはずの記録がない場合は、ブラウザを更新することをお勧めします。

次の図は、サーバーログのサンプルを示しています。

Logs myagent

---

Server Deployments

---

Showing page 1 of 2 Go to page:

Log Time	Level	Source	Service	Engine Name	Message
Jan 04, 2018 - 15:04:09.762	Info	Maintenance	Engine	myagent	Scheduled maintenance to perform database deletes has completed
Jan 04, 2018 - 15:04:09.762	Info	Maintenance	Engine	myagent	Scheduled maintenance has purged 0 server logs.
Jan 04, 2018 - 15:04:09.761	Info	Maintenance	Engine	myagent	Scheduled maintenance has purged 0 deployment logs.
Jan 04, 2018 - 15:04:09.761	Info	Maintenance	Engine	myagent	Scheduled maintenance has purged 24 offline contributions.
Jan 04, 2018 - 15:04:09.755	Info	System	Engine	myagent	Contribution 'project/rt.connector/PostTestServerConnectorICRT4096' with version number '1.0' (93) was deleted.
Jan 04, 2018 - 15:04:09.734	Info	System	Engine	myagent	Contribution 'project/rt.connector/ICRT4096GuideServiceCallConnection' with version number '1.0' (92) was deleted.
Jan 04, 2018 - 15:04:09.716	Info	System	Engine	myagent	Contribution 'project/spi/ICRT-3123-Agent' with version number '4.0' (60) was deleted.

このトピックでは次について説明します。

- サーバーログの情報
- プロセスサーバーログの削除

## サーバーログの情報

ログ情報が生成される際は、次のいずれかのフィルタが使用されています。

- **ログ時間（ログ後およびログ前）。**デフォルトでは、直近の 6 時間に発生したイベントがログに表示されます。最新の日時を表示するにはページを更新します。ログ期間は日時を設定することで変更できます。ショートカットとして、時間のテキストボックスの最後で **【今】** アイコンを選択することもできます。
- **レベル。**次のいずれかが選択されています。
  - 詳細—すべてのレベル
  - 情報
  - 警告
  - エラー
  - 重大
- **ソース。**イベントのイニシエータ、または **【すべて】** を選択します。ソースは次のとおりです。
  - システム—デプロイメントなどのプロセスサーバーのイベント
  - メンテナンス—ストレージおよびデプロイメントログの削除
  - ユーザー—ユーザーが開始
- **サービス。**サービスは次のとおりです。
  - 警告—警告サービスの設定イベント
  - 電子メール—電子メールサービスの設定イベント
  - エンジン—エンジンのイベント（デプロイメント、プロセスサーバーの開始/停止、ストレージイベント、監視の警告）
  - ID—ID サービス（ファイル、LDAP、JDBC）の設定イベント
  - メッセージ—メッセージサービス（JMS またはその他）の設定イベント
  - タスク—クライアント URI の更新イベント
  - プロセス—プロセスサーバーログコメントシステムサービスを使用して、プロセスサーバーログにプロセスから追加されるメッセージ
  - すべて—すべてのサービスが表示されます
- **エンジン名。**実行が発生したエンジン。

- **メッセージ。** イベントを説明するテキスト。

ページ要求またはスケジュール要求が発生すると、コンソールはユーザー/IP アドレスを記録します。アプリケーションサーバーには独自の方法があることがあります。例えば、次に示すのは Tomcat の場合の例です：

<https://tomcat.apache.org/tomcat-7.0-doc/api/org/apache/catalina/valves/RemoteIpValve.html>

## サーバーログの削除

【サーバーの環境設定】 ページの 【ストレージ】 タブにある 【ストレージのクリーンアップ】 を使用します。

さらに、サーバーログをクリアするためのインタフェースがシステムサービスとして Process Developer に用意されています。開発者はこれを使用してプロセスを作成およびデプロイし、プログラムを使用してログを削除できます。

## プロセスサーバーのログレベルの設定

プロセスサーバーのログを表示するには、次のフィルタを使用します。

- **最低レベル。** 次のいずれかの情報レベルを選択します。
  - 詳細 すべてのレベル
  - 情報
  - 警告
  - エラー
  - 重大
- **ロギングソース。** イベントのイニシエータ、または [すべて] を選択します。ソースは次のとおりです。
  - システム。デプロイメントなどのプロセスサーバーのイベント
  - メンテナンス。ストレージおよびデプロイメントログの削除
  - ユーザー。ユーザーが開始
- **ロギングサービス。** いずれかのサービスまたは [すべて] を選択します。サービスは次のとおりです。
  - 警告。警告サービスの設定イベント。
  - 電子メール。電子メールサービスの設定イベント。
  - エンジン。エンジンのイベント（デプロイメント、プロセスサーバーの開始/停止、ストレージイベント、監視の警告）
  - ID。ID サービス（ファイル、LDAP、JDBC）の設定イベント
  - メッセージ。メッセージサービス（JMS またはその他）の設定イベント。
  - タスク。クライアント URI の更新イベント。
  - プロセス。プロセスサーバーログコメントシステムサービスを使用して、プロセスサーバーログにプロセスから追加されるメッセージ。
- **ロギングエンジン名。** クラスタ内のいずれかのエンジンを選択するか、[すべて] を選択します。
- **ログ後およびログ前。** デフォルトでは、直近の 6 時間に発生したイベントがログに表示されます。最新の日時を表示するにはページを更新します。ログ期間は日時を設定することで変更できます。ショートカットとして、時間のテキストボックスの最後で [今] アイコンを選択することもできます。
- **メッセージ文字列。** ログのメッセージカラム内に、検索するテキスト文字列を入力します。メッセージを見つけるには、ワイルドカードを使用します。例えば、「Error」で始まるすべてのメッセージを検索するには、Error\*と入力します。「StorageException」という文字列を含むすべてのメッセージを検索するには、\*-StorageException\*と入力します。

## プロセスサーバーログの削除

プロセスサーバーログは、Secure Agent を選択した後、**【サーバーの環境設定】** > **【ストレージ】** から削除できます。

さらに、サーバーログをクリアするためのインターフェースがシステムサービスとして Process Developer に用意されています。開発者はこれを使用してプロセスを作成およびデプロイし、プログラムを使用してログを削除できます。