



Informatica®  
10.5

# Developer ワークフローガイド

本ソフトウェアおよびマニュアルは、使用および開示の制限を定めた個別の使用許諾契約のもとでのみ提供されています。本マニュアルのいかなる部分も、いかなる手段（電子的複製、写真複製、録音など）によっても、Informatica LLC の事前の承諾なしに複製または転載することは禁じられています。

米政府の権利プログラム、ソフトウェア、データベース、および関連文書や技術データは、米国政府の顧客に配信され、「商用コンピュータソフトウェア」または「商業技術データ」は、該当する連邦政府の取得規制と代理店固有の補足規定に基づきます。このように、使用、複製、開示、変更、および適応は、適用される政府の契約に規定されている制限およびライセンス条項に従うものとし、政府契約の条項によって適当な範囲において、FAR 52.227-19、商用コンピュータソフトウェアライセンスの追加権利を規定します。

Informatica および Informatica ロゴは、米国およびその他の国における Informatica LLC の商標または登録商標です。Informatica の商標の最新リストは、Web (<https://www.informatica.com/trademarks.html>) にあります。その他の企業名および製品名は、それぞれの企業の商標または登録商標です。

**本ソフトウェアまたはドキュメンテーション（あるいはその両方）の一部は、第三者が保有する著作権の対象となります。**必要な第三者の通知は、製品に含まれています。

本マニュアルの情報は、予告なしに変更されることがあります。このドキュメントで問題が見つかった場合は、[infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com) までご報告ください。

Informatica 製品は、それらが提供される契約の条件に従って保証されます。Informatica は、商品性、特定目的への適合性、非侵害性の保証等を含めて、明示的または黙示的ないかなる種類の保証をせず、本マニュアルの情報を「現状のまま」提供するものとします。

発行日: 2021-06-14

# 目次

<b>序文</b> .....	9
Informatica のリソース.....	9
Informatica Network.....	9
Informatica ナレッジベース.....	9
Informatica マニュアル.....	9
Informatica 製品可用性マトリックス.....	10
Informatica Velocity.....	10
Informatica Marketplace.....	10
Informatica グローバルカスタマサポート.....	10
 <b>第 1 章: ワークフロー</b> .....	11
ワークフローの概要.....	11
ワークフローの開発.....	12
ワークフローオブジェクト.....	12
イベント.....	13
タスク.....	13
ゲートウェイ.....	14
シーケンスフロー.....	14
条件付きシーケンスフロー.....	15
条件付きシーケンスフローのパラメータおよび変数.....	16
ワークフローの作成.....	16
ワークフローへのオブジェクトの追加.....	17
オブジェクトの接続.....	17
条件付きシーケンスフローの作成.....	17
ワークフローの検証.....	18
ワークフローオブジェクトの検証.....	18
シーケンスフローの検証.....	18
式の検証.....	19
ワークフローの検証.....	19
PowerCenter からのワークフローのインポート.....	19
PowerCenter からワークフローをインポートするためのルールとガイドライン.....	20
PowerCenter からのコマンドタスクのインポート.....	20
コマンドタスクのインポートに伴う変換動作.....	21
ワークフローの詳細プロパティ.....	21
ワークフローのデプロイメント.....	23
ワークフローのデプロイと実行.....	23
ワークフローの実行.....	23
ワークフローの監視.....	24
ワークフローの削除.....	24
ワークフローの例.....	24

例: マッピングの実行前と実行後のコマンドの実行. . . . .	24
例: 複数のシーケンスフローの作成. . . . .	25
<b>第2章: ワークフロー変数. . . . .</b>	<b>27</b>
ワークフロー変数の概要. . . . .	27
タスク入力. . . . .	28
タスク出力. . . . .	28
システムワークフロー変数. . . . .	29
ユーザー定義ワークフロー変数. . . . .	29
ユーザー定義変数の作成. . . . .	29
割り当てタスクを使用した値の割り当て. . . . .	30
タスク出力を使用した値の割り当て. . . . .	31
ワークフロー変数の使用場所. . . . .	32
タスク入力への変数の割り当て. . . . .	33
式および文字列の変数名. . . . .	34
文字列内のエスケープ文字. . . . .	34
ネストされた変数. . . . .	35
ワークフロー変数のデータ型変換. . . . .	36
日付変数の形式の変更. . . . .	36
<b>第3章: ワークフローパラメータ. . . . .</b>	<b>38</b>
ワークフローパラメータの概要. . . . .	38
タスク入力. . . . .	39
パラメータを使用してワークフローを実行するプロセス. . . . .	39
ワークフローパラメータの使用場所. . . . .	39
タスク入力へのワークフローパラメータの割り当て. . . . .	40
式および文字列のパラメータ名. . . . .	41
文字列のエスケープ文字. . . . .	41
ネストされたパラメータ. . . . .	42
ユーザー定義マッピングパラメータのワークフローパラメータの作成. . . . .	42
デフォルトのパラメータ値. . . . .	43
ワークフローパラメータのデータ型変換. . . . .	44
パラメータセット. . . . .	44
パラメータセットの作成. . . . .	46
パラメータセットを使用したワークフローの実行. . . . .	48
パラメータファイル. . . . .	48
ワークフローのパラメータファイルの例. . . . .	48
パラメータファイルの例. . . . .	49
パラメータファイルに関するルールとガイドライン. . . . .	49
パラメータファイルのエクスポート. . . . .	50
infacmd ms ListMappingParams からのパラメータファイルの作成. . . . .	50
パラメータファイルを使用したワークフローの実行. . . . .	51

<b>第 4 章 : クラスタタスク</b>	<b>52</b>
クラスタタスクの概要	52
クラスタの作成タスク	52
クラスタの作成タスクの全般プロパティ	53
クラスタの作成タスクの出力	53
Amazon EMR の詳細プロパティ	53
Azure Databricks の詳細プロパティ	56
AWS Databricks の詳細プロパティ	58
クラスタの削除タスク	61
<b>第 5 章 : コマンドタスク</b>	<b>62</b>
コマンドタスクの概要	62
コマンドの構文	62
コマンドのパラメータおよび変数	63
コマンドタスクの入力	63
コマンドタスクの出力	64
コマンドタスクの詳細プロパティ	65
コマンドタスクのトラブルシューティング	65
<b>第 6 章 : ヒューマンタスク</b>	<b>66</b>
ヒューマンタスクの概要	66
ヒューマンタスクと例外データ管理	67
例外データのタイプ	67
Analyst ツール	67
ヒューマンタスクのインスタンス	68
ヒューマンタスクのステップ	68
ヒューマンタスクのロール	69
ヒューマンタスクのプロパティ	69
[全般] タブ	70
[パーティシパント] タブ	70
[データソース] タブ	71
[タスク分散] タブ	72
[通知] タブ	72
[入力] タブ	73
[出力] タブ	73
[詳細] タブ	74
ステップのプロパティ	74
全般オプション	74
設定オプション	75
パーティシパントオプション	75
権限オプション	76
タイムアウトオプション	77

通知オプション.....	77
ヒューマンタスクとワークフローの設定.....	79
ヒューマンタスクの設定.....	79
ヒューマンタスクのタスクインスタンスの設定.....	79
タスクステップの設定.....	80
タスクインスタンスの作成.....	80
同じサイズのタスクインスタンスの作成.....	81
データ値に基づくタスクインスタンスの作成.....	81
ステップに対する権限の設定.....	82

## 第7章: マッピングタスク..... 84

マッピングタスクの概要.....	84
同じマッピングを実行する複数のマッピングタスク.....	84
マッピングタスクの全般プロパティ.....	85
マッピングタスクの入力.....	86
マッピングパラメータとマッピングタスク.....	87
ワークフローの実行時のマッピングパラメータのオーバーライド.....	87
マッピングタスクでのパラメータの使用.....	88
マッピング入力パラメータ値の自動入力.....	89
割り当てのクリア.....	89
マッピングタスクの出力.....	90
システム定義のマッピング出力.....	90
マッピングの保持出力.....	91
マッピング出力のワークフロー変数へのバインド.....	93
マッピング出力の設定方法.....	95
マッピングタスクログ.....	103
マッピングタスクログファイル名.....	104
マッピングタスクログファイルディレクトリ.....	104
マッピングタスクの詳細プロパティ.....	104
カスタムプロパティ.....	107
設定プロパティの変数およびパラメータの値.....	108
タスクで実行するマッピングの変更.....	110
クラスタワークフロー内のマッピングタスク.....	110

## 第8章: 通知タスク..... 111

通知タスクの概要.....	111
受信者.....	112
受信者の選択.....	112
受信者の通知タイプの設定.....	113
受信者名の入力.....	113
動的な電子メール受信者.....	113
電子メールアドレス.....	114
電子メールアドレスの入力.....	115

動的な電子メールアドレス.....	115
電子メールのコンテンツ.....	116
動的な電子メールのコンテンツ.....	116
電子メールのコンテンツの入力.....	117
通知タスクの出力.....	117
通知タスクの詳細プロパティ.....	118
通知タスクのトラブルシューティング.....	118
<b>第 9 章: ゲートウェイ.....</b>	<b>119</b>
ゲートウェイの概要.....	119
排他ゲートウェイ.....	120
排他ゲートウェイの例.....	120
包含ゲートウェイ.....	121
包含ゲートウェイの例.....	122
デフォルトシーケンスフロー.....	123
シーケンスフローの複数のブランチへの分割.....	123
ブランチの 1 つのシーケンスフローへのマージ.....	124
<b>第 10 章: ワークフローリカバリ.....</b>	<b>125</b>
ワークフローリカバリの概要.....	125
リカバリ可能なワークフローの状態.....	126
リカバリ不能なワークフローの状態.....	126
タスクのエラーおよびリカバリ.....	126
ワークフローリカバリストラテジ.....	127
ワークフローリカバリと終結イベント.....	128
グリッドでのワークフローリカバリ.....	128
タスクリカバリストラテジ.....	129
リカバリ動作の再開.....	130
スキップリカバリの動作.....	131
ヒューマンタスクの再開動作.....	132
ワークフローおよびタスクリカバリのルールとガイドライン.....	132
リカバリの設定手順.....	133
ワークフローのリカバリの設定.....	133
タスクリカバリストラテジの設定.....	133
ワークフローリカバリ中のパラメータと変数の値.....	134
ワークフローリカバリログ.....	135
再開のマッピングの開発手順.....	135
手動でのターゲット行の削除.....	135
ターゲット行を削除するマッピングの設定.....	135
ワークフローインスタンスのリカバリ.....	137
中断後のワークフローの状態のサマリ.....	137
ワークフローリカバリの例.....	138

<b>第 11 章 : ワークフロー管理</b>	<b>141</b>
ワークフロー管理の概要	141
ワークフローグラフ	142
ワークフローのグラフの表示	142
ワークフローオブジェクトの表示	142
ワークフローオブジェクトのサマリ統計の表示	143
ワークフローオブジェクトの詳細統計の表示	143
ワークフローの状態	144
ワークフローオブジェクトの状態	145
マッピングタスクの作業項目の状態	146
ワークフローのキャンセルまたは強制終了	146
ワークフローリカバリ	147
リカバリプロパティ	148
ワークフローのリカバリ	148
ワークフローログ	148
ワークフローログ情報	149
ワークフローのログ表示	149
ワークフロー内のマッピング実行のログ表示	149
<b>索引</b>	<b>150</b>



# 序文

『*Informatica® Developer* ワークフローガイド』を使用して、ワークフローの作成、実行、管理方法を学びます。ワークフローの概念を理解して、マッピングとその他のタスクを単一の操作で実行できるようにします。中断時やエラー発生時のワークフローのリカバリ方法を学びます。

## Informatica のリソース

Informatica は、Informatica Network やその他のオンラインポータルを通じてさまざまな製品リソースを提供しています。リソースを使用して Informatica 製品とソリューションを最大限に活用し、その他の Informatica ユーザーや各分野の専門家から知見を得ることができます。

### Informatica Network

Informatica Network は、Informatica ナレッジベースや Informatica グローバルカスタマサポートなど、多くのリソースへの入口です。Informatica Network を利用するには、<https://network.informatica.com> にアクセスしてください。

Informatica Network メンバーは、次のオプションを利用できます。

- ナレッジベースで製品リソースを検索できます。
- 製品の提供情報を表示できます。
- サポートケースを作成して確認できます。
- 最寄りの Informatica ユーザーグループネットワークを検索して、他のユーザーと共同作業を行えます。

### Informatica ナレッジベース

Informatica ナレッジベースを使用して、ハウツー記事、ベストプラクティス、よくある質問に対する回答など、製品リソースを見つけることができます。

ナレッジベースを検索するには、<https://search.informatica.com> にアクセスしてください。ナレッジベースに関する質問、コメント、ご意見の連絡先は、Informatica ナレッジベースチーム ([KB\\_Feedback@informatica.com](mailto:KB_Feedback@informatica.com)) です。

### Informatica マニュアル

Informatica マニュアルポータルでは、最新および最近の製品リリースに関するドキュメントの膨大なライブラリを参照できます。マニュアルポータルを利用するには、<https://docs.informatica.com> にアクセスしてください。

製品マニュアルに関する質問、コメント、ご意見については、Informatica マニュアルチーム ([infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com)) までご連絡ください。

## Informatica 製品可用性マトリックス

製品可用性マトリックス (PAM) には、製品リリースでサポートされるオペレーティングシステム、データベースなどのデータソースおよびターゲットが示されています。Informatica PAM は、<https://network.informatica.com/community/informatica-network/product-availability-matrices> で参照できます。

## Informatica Velocity

Informatica Velocity は、Informatica プロフェッショナルサービスが開発したヒントとベストプラクティスのコレクションで、多数のデータ管理プロジェクトから得た実体験に基づいています。Informatica Velocity には、世界中の組織と連携してデータ管理ソリューションを計画、開発、デプロイ、管理する Informatica コンサルタントによる集合知を表しています。

Informatica Velocity リソースには、<http://velocity.informatica.com> からアクセスしてください。Informatica Velocity についての質問、コメント、またはアイデアがある場合は、[ips@informatica.com](mailto:ips@informatica.com) から Informatica プロフェッショナルサービスにお問い合わせください。

## Informatica Marketplace

Informatica Marketplace は、お使いの Informatica 製品を拡張したり強化したりするソリューションを検索できるフォーラムです。Marketplace で、Informatica デベロッパーやパートナーからの多数のソリューションを活用すれば、生産性を向上したり、プロジェクトでの実装時間を短縮したりできます。Informatica Marketplace は、<https://marketplace.informatica.com> からアクセスしてください。

## Informatica グローバルカスタマサポート

電話または Informatica Network を介してグローバルカスタマサポートに連絡できます。

各地域の Informatica グローバルカスタマサポートの電話番号は、Informatica Web サイト (<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>) を参照してください。

Informatica Network のオンラインサポートリソースを見つけるには、<https://network.informatica.com> にアクセスして eSupport オプションを選択します。

# 第 1 章

## ワークフロー

この章では、以下の項目について説明します。

- [ワークフローの概要, 11 ページ](#)
- [ワークフローの開発, 12 ページ](#)
- [ワークフローオブジェクト, 12 ページ](#)
- [シーケンスフロー, 14 ページ](#)
- [ワークフローの作成, 16 ページ](#)
- [ワークフローへのオブジェクトの追加, 17 ページ](#)
- [オブジェクトの接続, 17 ページ](#)
- [条件付きシーケンスフローの作成, 17 ページ](#)
- [ワークフローの検証, 18 ページ](#)
- [PowerCenter からのワークフローのインポート, 19 ページ](#)
- [ワークフローの詳細プロパティ, 21 ページ](#)
- [ワークフローのデプロイメント, 23 ページ](#)
- [ワークフローの実行, 23 ページ](#)
- [ワークフローの監視, 24 ページ](#)
- [ワークフローの削除, 24 ページ](#)
- [ワークフローの例, 24 ページ](#)

## ワークフローの概要

ワークフローは、ビジネスプロセスを定義する一連のイベント、タスク、およびディシジョンを視覚的に表したものです。Developer tool を使用してワークフローを作成し、そのワークフローをモデルリポジトリに保存します。ワークフローをアプリケーションとしてデータ統合サービスにデプロイします。

Developer tool でワークフローを作成します。ワークフローにオブジェクトを追加し、シーケンスフローでそのオブジェクトを接続します。データ統合サービスはワークフローで設定した手順を使用してオブジェクトを実行します。

ワークフローオブジェクトには、イベント、タスク、ゲートウェイがあります。イベントは、ワークフローを開始または終了します。タスクは、マッピングの実行、電子メールの送信、シェルコマンドの実行など、ワークフローの 1 つの作業単位を実行するアクティビティです。ゲートウェイでは、ワークフローでデータパスを分割するか、マージするかの決定を定義します。

シーケンスフローは、あるワークフローオブジェクトを別のワークフローオブジェクトに接続します。シーケンスフローは、データ統合サービスがオブジェクトを実行する順序を指定します。シーケンスフローに対する条件を定義して、そのシーケンスフローが指定するオブジェクトをデータ統合サービスが実行するかどうかを決定します。

ワークフローの変数やパラメータを定義して使用すると、ワークフローの柔軟性を高めることができます。ワークフロー変数は、ランタイム情報を記録する、ワークフローの実行時に変更できる値を表します。ワークフローパラメータは、ワークフローの実行前に定義する定数値を表します。ワークフローの変数とパラメータは、条件付きシーケンスフローやオブジェクトのフィールドで使用できます。また、タスクとワークフローの間でデータをやり取りする場合にも使用します。

ワークフローのリカバリを設定して、中断されたワークフローインスタンスを完了できます。エラーが発生した場合、ワークフローインスタンスをキャンセルした場合、またはサービスプロセスが予期せずにシャットダウンした場合は、実行中のワークフローインスタンスが中断されることがあります。強制終了されたワークフローはリカバリできません。

ワークフローをデプロイしたら、`infacmd wfs` コマンドラインプログラムを使用して、デプロイ済みのアプリケーションからワークフローのインスタンスを実行します。ワークフローインスタンスの実行は Monitoring ツールで監視します。

## ワークフローの開発

ワークフローを開発するには、ワークフローで実行するオブジェクトを選択し、オブジェクトをシーケンスフローと接続します。定義した条件に基づいて、ゲートウェイを使用してシーケンスフローを分割またはマージできます。

1. ワークフローを作成します。
2. ワークフローにオブジェクトを追加し、オブジェクトのプロパティを設定します。
3. シーケンスフローでオブジェクトを接続して、オブジェクトの実行順序を指定します。
4. ランタイム情報を取得するワークフローの変数を定義します。ワークフロー変数は、条件付きシーケンスフローやオブジェクトのフィールドで使用します。
5. ワークフローを実行するたびにパラメータ値を変更できるように、ワークフローのパラメータを定義します。ワークフローパラメータは、条件付きシーケンスフローやオブジェクトのフィールドで使用します。
6. 必要に応じて、ワークフローのリカバリを設定します。
7. ワークフローに含まれている可能性のあるエラーを識別するには、ワークフローを検証します。検出したエラーを修正します。
8. ワークフローをアプリケーションに追加し、そのアプリケーションをデータ統合サービスにデプロイします。

ワークフローをデプロイしたら、`infacmd wfs` コマンドラインプログラムを使用して、デプロイ済みのアプリケーションからワークフローのインスタンスを実行します。ワークフローインスタンスの実行は Monitoring ツールで監視します。

## ワークフローオブジェクト

ワークフローオブジェクトには、イベント、タスク、ゲートウェイがあります。オブジェクトは、エディタでワークフローを開発するときに追加します。ワークフローオブジェクトは再利用することはできず、ワークフロー内にものみ保存されます。

## イベント

イベントは、ワークフローでデータに指定されるアクティビティの開始と終了を定義します。各ワークフローには開始イベントと終了イベントがあります。必要に応じて、ワークフローに終結イベントを1つ以上含めることができます。

イベントには、Start\_Event、End\_Event、または Terminate\_Event というデフォルトの名前が付けられます。イベントのプロパティで名前を変更したり説明を追加したりできます。

次の表に、ワークフローに追加できるすべてのイベントを示します。

イベント	説明
開始	ワークフローの先頭を定義します。開始イベントは、ワークフローでデータに指定される可能なアクションのシーケンスの開始を表します。ワークフローには1つの開始イベントが含まれます。
終了	ワークフローの末尾を定義します。終了イベントは、ワークフローでデータに指定される可能なアクションのシーケンスの完了を表します。ワークフローには1つの終了イベントが含まれます。
終結	終了イベントより前の、ワークフローを終了できるポイントを定義します。タスクを終結イベントに接続し、タスクの出力がシーケンスフローの条件を満たす場合、ワークフローが終了します。終結イベントは、ワークフローの後続タスクが実行される前にワークフローを強制終了します。ワークフローに終結イベントを1つ以上含めることができます。

**注:** ヒューマンタスクにも開始イベントと終了イベントが含まれます。これらのイベントによって、1つ以上のステップのシーケンスフローがヒューマンタスクに作成されます。ヒューマンタスクに終結イベントを含めることはできません。

## タスク

タスクは、マッピングの実行、電子メールの送信、シェルコマンドの実行など、ワークフローの1つの作業単位を実行するアクティビティです。タスクは、ワークフローの実行時に実行される処理を表します。エディタでは四角形で示されます。

次の表に、ワークフローに追加できるすべてのタスクを示します。

タスク	説明
割付	ユーザー定義ワークフロー変数に値を割り当てます。
コマンド	1つのシェルコマンドを実行するか、外部の実行可能プログラムを起動します。
ヒューマン	ワークフローデータで1人以上のユーザーが実行するアクションを定義します。Analyst ツールユーザーにマッピングタスクで実行するマッピングの出力の分析をさせる場合に作成します。
マッピング	マッピングを実行します。
通知	指定した受信者に電子メール通知を送信します。
投票	アクティブではありません。Analyst ツールユーザーは承認ワークフローを実行し、ビジネス用語集のアセットを承認または拒否します。Analyst ツールユーザーは、承認ワークフローの設定や実行のために Developer tool のワークフロー機能と対話することはありません。

同じタイプの複数のタスクをワークフローに含めることができます。

タスクには、<task type>\_Task という形式のデフォルトの名前が付けられます (Command\_Task など)。同じワークフローに同じタイプの別のタスクを追加すると、デフォルトの名前に整数が追加されます (Command\_Task1 など)。タスクの全般プロパティで名前を変更したり説明を追加したりできます。

## ゲートウェイ

ゲートウェイは 1 つのシーケンスフローを複数のブランチに分割したり、複数のブランチを 1 つのシーケンスフローにマージしたりします。ゲートウェイはペアで機能します。一方のゲートウェイでは、ワークフローデータがたどることができるブランチを定義します。他方のゲートウェイでは、ブランチを 1 つのシーケンスフローにリストアします。各ブランチを開始するシーケンスフローに条件を追加できます。データ統合サービスは、条件値を使用してワークフローデータがたどるブランチを識別します。

### 排他ゲートウェイと包含ゲートウェイ

排他ゲートウェイと包含ゲートウェイを 1 つのワークフローに追加できます。ゲートウェイをペアでワークフローに追加します。ゲートウェイの間の 1 つのブランチにワークフローデータが進む必要がある場合は、排他ゲートウェイを追加します。ゲートウェイの間の複数のブランチにワークフローデータが進む可能性がある場合は、包含ゲートウェイを追加します。包含ゲートウェイを追加すると、データ統合サービスは各ブランチで並行してオブジェクトを実行します。

Developer tool はゲートウェイに Exclusive\_Gateway または Inclusive\_Gateway というデフォルト名を設定します。同じワークフローに別の排他ゲートウェイを追加すると、デフォルトの名前に整数が追加されます (Exclusive\_Gateway1 など)。ゲートウェイプロパティで名前の変更や説明の追加ができます。

#### 関連項目：

- [「排他ゲートウェイ」 \(ページ 120\)](#)
- [「包含ゲートウェイ」 \(ページ 121\)](#)

## シーケンスフロー

シーケンスフローは、ワークフローオブジェクトを接続するコネクタであり、オブジェクトの実行順序を示します。エディタでは矢印で示されます。次のオブジェクトを実行するかどうかを判定するための条件付きシーケンスフローを作成することができます。

シーケンスフローを使用してループを作成することはできません。各シーケンスフローは 1 回ずつ実行できます。

オブジェクトに含めることができる入力シーケンスフローおよび出力シーケンスフローの数は、オブジェクトのタイプによって異なります。

#### イベント

開始イベントには 1 つの出力シーケンスフローが必要です。終了イベントには 1 つの入力シーケンスフローが必要です。

#### タスク

タスクには、入力シーケンスフローと出力シーケンスフローが 1 つずつ必要です。

## ゲートウェイ

ゲートウェイには、複数の入力シーケンスフローか複数の出力シーケンスフローのどちらかが必要です。それらを両方含めることはできません。ワークフローを分割するには、ゲートウェイからの出力シーケンスフローを複数使用します。複数のブランチを 1 つのフローにマージする場合は、ゲートウェイへの入力シーケンスフローを複数使用します。

オブジェクトを接続すると、シーケンスフローにデフォルトの名前が付けられます。シーケンスフローの名前の形式は次のとおりです。

<originating object name>\_to\_<ending object name>

条件付きシーケンスフローを作成する場合は、シーケンスフローの名前を条件式を示す名前に変更すると便利です。例えば、マッピングタスクからコマンドタスクへの条件付きシーケンスフローに、マッピングタスクが正常に実行されたかどうかを確認する条件が含まれている場合は、シーケンスフローの名前を MappingSucceeded などに変更します。シーケンスフローの全般プロパティで名前を変更したり説明を追加したりできます。

## 条件付きシーケンスフロー

ワークフローの次のオブジェクトを実行するかどうかを判定するための条件付きシーケンスフローを作成します。

条件付きシーケンスフローには、true または false に評価される式が含まれます。式の戻り値は、論理値または整数値でなければなりません。式の戻り値が整数値の場合、0 以外の値はすべて true とみなされ、0 は false とみなされます。

式が true と評価された場合、次のオブジェクトが実行されます。式が false と評価された場合は、次のオブジェクトは実行されません。シーケンスフローで条件を指定しなかった場合は、デフォルトで次のオブジェクトが実行されます。

条件付きシーケンスフローの式が false と評価された場合、次のオブジェクトやそのブランチに含まれる以降のオブジェクトは実行されません。Monitoring ツールでワークフローを監視する場合、ワークフローで実行されないオブジェクトは表示されません。実行されないオブジェクトが含まれている場合でも、ワークフローは正常に完了できます。

開始イベントからワークフローの次のオブジェクトへの条件付きシーケンスフロー、またはワークフローの最後のオブジェクトから終了イベントへの条件付きシーケンスフローを作成することはできません。

## 失敗したタスクと条件付きシーケンスフロー

デフォルトでは、データ統合サービスはタスクが失敗しても引き続きワークフロー内の後続のオブジェクトを実行します。タスクが失敗したときに以降のワークフローオブジェクトの実行を中止するには、前のタスクが成功したかどうかを確認する条件付きシーケンスフローを使用します。

条件付きシーケンスフローを使用すると、マッピング、コマンド、通知、またはヒューマンタスクが成功したかチェックすることができます。これらのタスクは全般出力に「成功」を返します。「成功」出力には、タスクが正常に実行された場合は true が、タスクが失敗した場合は false が含まれています。まず、タスクが返す「成功」出力を取得するためのワークフローブール変数を作成します。次に、出力の条件付きシーケンスフローで、変数が true かどうかを確認する式を作成します。

例えば、マッピングタスクが返す「成功」出力を取得するためのワークフローブール変数を作成します。そして条件付きシーケンスフローに次の式を作成すると、これによりマッピングタスクの後にワークフロー内の次のタスクが実行されるようになります。

```
$var:MappingTaskSuccessful = true
```

マッピングタスクが失敗した場合、この式で失敗が評価され、データ統合サービスが以降のワークフローオブジェクトの実行を停止します。

#### 関連項目：

- [「タスクリカバリストラテジ」 \(ページ 129\)](#)
- [「タスク出力」 \(ページ 28\)](#)

### 終結イベントと条件付きシーケンスフロー

条件付きシーケンスフローを使用して、タスクを終結イベントに接続できます。タスクの出力がシーケンスフローの条件を満たす場合、ワークフローが終結イベントに到達し、ワークフローが終了します。終結イベントで終了したワークフローは強制終了状態に入ります。

## 条件付きシーケンスフローのパラメータおよび変数

条件付きシーケンスフローの式には、ワークフローのパラメータや変数を含めることができます。

ワークフローのパラメータや変数は、**【条件】** タブで選択できます。また、所定の構文に従って、条件式の中にパラメータ名または変数名を入力することもできます。

例えば、マッピングタスクで実行されるマッピングによってターゲットに書き込まれる行数を取得するワークフロー変数を作成し、マッピングタスクをコマンドタスクに接続する条件付きシーケンスフローで次の式を作成します。

```
$var:TargetRowsMapping > 500
```

これにより、マッピングによってターゲットに書き込まれた行数が 500 行を超えるとコマンドタスクが実行されるようになります。

#### 関連項目：

- [「式および文字列のパラメータ名」 \(ページ 41\)](#)
- [「式および文字列の変数名」 \(ページ 34\)](#)

## ワークフローの作成

ワークフローを作成すると、開始イベントと終了イベントがワークフローに追加されます。

1. **Object Explorer** ビューで、プロジェクトまたはフォルダを選択します。
2. **【ファイル】 > 【新規】 > 【ワークフロー】** をクリックします。  
ワークフローにはデフォルトの名前が付けられます。
3. 必要に応じて、デフォルトのワークフロー名を編集します。
4. **【完了】** をクリックします。  
開始イベントと終了イベントを含むワークフローがエディタに表示されます。



## ワークフローへのオブジェクトの追加

ワークフローで実行するタスクやゲートウェイを追加します。ワークフローには、開始イベントと終了イベントが1つずつ必要です。ワークフローを作成すると、開始イベントと終了イベントがワークフローに追加されます。

1. エディタでワークフローを開きます。
2. **ワークフローオブジェクト**パレットでオブジェクトを選択し、エディタにドラッグします。マッピングタスクを選択した場合は、**[参照]** をクリックしてマッピングを選択し、**[完了]** をクリックします。  
マッピングタスクを追加する場合は、**Object Explorer** ビューでマッピングを選択し、エディタにドラッグします。  
オブジェクトがエディタに表示されます。プロパティを設定するオブジェクトを選択します。

## オブジェクトの接続

シーケンスフローでオブジェクトを接続して、ワークフローのオブジェクトの実行順序を指定します。

2つのオブジェクトを接続するには、エディタで1つ目のオブジェクトを選択し、2つ目のオブジェクトにドラッグします。複数のオブジェクトを接続するには、**[ワークフローオブジェクトの接続]** ダイアログボックスを使用します。

1. エディタで右クリックし、**[ワークフローオブジェクトの接続]** を選択します。  
**[ワークフローオブジェクトの接続]** ダイアログボックスが表示されます。
2. 接続元のオブジェクト、接続先のオブジェクトの順に選択し、**[適用]** をクリックします。
3. 他のオブジェクトについても同様に接続し、終了したら **[OK]** をクリックします。  
オブジェクト間のシーケンスフローが表示されます。

## 条件付きシーケンスフローの作成

条件付きシーケンスフローには、true または false に評価される式が含まれます。ワークフローの次のオブジェクトを実行するかどうかを判定するための条件付きシーケンスフローを作成します。

1. エディタでシーケンスフローを選択します。
2. **[プロパティ]** ビューで **[条件]** タブをクリックします。
3. 条件式を入力します。  
**[関数]** タブにトランスフォーメーション言語関数が表示され、**[入力]** タブにワークフローのパラメータと変数が表示されます。関数、パラメータ、または変数の名前をダブルクリックして式に含めます。  
必要に応じて、演算子およびリテラル値を式に入力します。
4. **[検証]** ボタンを使用して条件を検証します。  
ダイアログボックスにエラーが表示されます。
5. エラーが表示された場合は、エラーを修正してから再度検証します。

# ワークフローの検証

ワークフローを開発するときは、Data Integration Service でワークフロー全体を読み取って処理できるように設定する必要があります。Data Integration Service でのワークフローの実行を妨げるエラーが Developer ツールで検出された場合、そのワークフローは無効とマークされます。

Developer ツールでのワークフローの検証では、シーケンスフロー、式、およびワークフローオブジェクトが検証されます。

## ワークフローオブジェクトの検証

ワークフローオブジェクトの検証は、Developer ツールでワークフローを検証するときに毎回実行されます。

Developer ツールでは、以下のワークフローオブジェクトが検証されます。

### イベント

ワークフローには、1つの開始イベントが最初のオブジェクトとして含まれ、1つの終了イベントが最後のオブジェクトとして含まれている必要があります。また、開始イベントから終了イベントまでのパスが必要です。

### タスク

各タスクの名前はワークフロー内で一意である必要があります。ワークフローのパラメータおよび変数にタスク入力割り当てられている場合や、ワークフロー変数にタスク出力割り当てられている場合は、互換性があるデータ型でなければなりません。タスクの設定プロパティは有効な値に割り当てられている必要があります。

各割り当てタスクでは、1つのワークフロー変数に有効な値を割り当てる必要があります。ワークフロー変数に割り当てられる値は、互換性があるデータ型でなければなりません。タスクの割り当て式でワークフローのパラメータや変数が使用されている場合は、それらのパラメータと変数が存在することが確認されます。

各コマンドタスクに、有効なコマンドが必要です。コマンドに改行文字や復帰文字が含まれてはなりません。コマンドでワークフローのパラメータや変数が使用されている場合は、それらのパラメータと変数が存在することが確認されます。

各マッピングタスクに、リポジトリに存在する有効なマッピングが必要です。

各通知タスクに、受信者が少なくとも1人含まれている必要があります。タスクでワークフローのパラメータや変数が使用されている場合は、それらのパラメータと変数が存在することが確認されます。

### ゲートウェイ

各ゲートウェイの名前はワークフロー内で一意である必要があります。

## シーケンスフローの検証

シーケンスフローの検証は、Developer ツールでワークフローを検証するときに毎回実行されます。

Developer ツールによるシーケンスフローの検証では、次の点が検証されます。

- ワークフローには、ループしたシーケンスフローを含めないでください。各シーケンスフローは1回ずつ実行できます。
- 開始イベントには出力シーケンスフローが1つあります。
- 開始イベントからのシーケンスフローには条件が含まれません。
- 終了イベントには入力シーケンスフローが1つあります。
- 各タスクには、入力シーケンスフローと出力シーケンスフローが1つずつ含まれます。

- 各ゲートウェイには、複数の入力シーケンスフローが複数の出力シーケンスフローのどちらかが必要ですが、それらを両方含めることはできません。ワークフローを分割するゲートウェイには、出力シーケンスフローが少なくとも2つ含まれており、いずれかがデフォルトとして設定されている必要があります。ワークフローをマージするゲートウェイに、デフォルトの出力シーケンスフローを含めることはできません。
- 条件付きシーケンスフローの式は、ブール値または整数値を返す必要があります。式に改行文字や復帰文字を含めることはできません。

## 式の検証

条件付きシーケンスフローまたは割り当てタスクの式の検証は、式を作成するときに実行できます。エラーを修正しなかった場合、ワークフローの検証時に**検証ログ**ビューにエラーメッセージが表示されます。

## ワークフローの検証

ワークフローを検証して、Data Integration Service でワークフロー全体を読み取って処理できることを確認します。

1. エディタでワークフローを開きます。
2. **【編集】 > 【検証】** をクリックします。  
エラーは**検証ログ**ビューに表示されます。
3. エラーが表示された場合は、エラーを修正してから再度検証します。

# PowerCenter からのワークフローのインポート

PowerCenter からワークフローをインポートすると、開始タスクおよびセッションタスク（再利用可能なタスクと再利用可能でないタスクの両方を含む）がモデルリポジトリにインポートされます。モデルリポジトリでは、PowerCenter からインポートした後に、各ワークフローに終了イベントが含まれます。

Developer tool または infacmd を使用して PowerCenter からマッピングをインポートするか、マッピングを PowerCenter にエクスポートしてメタデータを再利用します。

PowerCenter からモデルリポジトリにデータをインポートするには、以下の作業を完了します。

1. PowerCenter Client または以下のコマンドを使用して、PowerCenter オブジェクトをファイルにエクスポートします。  
`pmrep ExportObject`
2. 以下のコマンドを使用して、エクスポートファイルをモデルリポジトリファイルに変換します。  
`infacmd ipc importFromPC`
3. Developer tool または以下のコマンドを使用して、オブジェクトをインポートします。  
`infacmd tools importObjects`

PowerCenter ワークフロー内のブランチは、モデルリポジトリ内のインポート済みのワークフローの包含ゲートウェイを経由してルーティングされます。包含ゲートウェイには、デフォルトで、少なくとも1つの出力シーケンスフローが設定されている必要があります。ゲートウェイからの最初の出力シーケンスフローがデフォルトになり、インポート済みのワークフローは有効のままとなります。

## PowerCenter からワークフローをインポートするためのルールとガイドライン

ワークフローをインポートする場合、次のルールとガイドラインを考慮してください。

**インポートプロセスは、出力リンクのない PowerCenter セッションまたはタスクインスタンスを、モデルリポジトリ内の終了イベントに関連付けます。**

PowerCenter ワークフローに、出力リンクのないセッションまたはタスクが含まれている場合、インポート済みのワークフローにはゲートウェイへのリンクが追加され、セッションまたはタスクインスタンスがモデルリポジトリの終了イベントに関連付けられます。

**インポート済みのワークフローは、サポートされていない PowerCenter タスクを削除します。ただし、サポートされていないタスクがブランチまたはマージの一部である場合、サポートされていないタスクは削除され、ゲートウェイタスクのあるブランチまたはマージに置き換えられます。**

PowerCenter マッピングのワークフローに、モデルリポジトリへのインポートがサポートされていないタスクが含まれている場合、サポートされていないタスクは、モデルリポジトリ内のインポート済みのワークフローから削除されます。ただし、サポートされていないタスクがブランチまたはマージの一部である場合、これらのタスク用のゲートウェイが作成され、最終的にインポートされるワークフローでは、関連付けられていないタスクとして表示されます。

**インポートプロセスによってセッションタスクは分割され、ターゲットのロード順に基づいて各パイプラインに対応するマッピングごとにセッションタスクが追加されます。**

ワークフローに PowerCenter のセッションタスクが含まれている場合、モデルリポジトリにインポートした後、セッションが、各ターゲットのロード順に対して複数のパイプラインのあるマッピングを指定すると、タスクは分割されます。インポート済みのワークフローには、ターゲットのロード順に基づいて各パイプラインに対応するマッピングごとのセッションタスクが追加されます。

例えば、セッションに関連付けられた開始タスクを含む PowerCenter ワークフローがあるとします。セッションは、2つのパイプラインを持つマッピングを指定しています。最初のパイプラインには、ソース Source1 があり、トランスフォーメーション Tx1 を指しています。これが、ターゲット Target1 に接続されます。2番目のパイプラインには、ソース Source2 があり、トランスフォーメーション Tx2 を指しています。これが、ターゲット Target2 に接続されます。

モデルリポジトリにワークフローをインポートすると、次のように2つのセッションが作成され、ターゲットのロード順に従って関連付けられます。開始 -> Session\_Mapping\_pipeline\_1-> Session\_Mapping\_pipeline\_2-> 終了。ここで Session\_Mapping\_pipeline\_1 は最初のパイプラインから作成されたマッピングを指し、Session\_Mapping\_pipeline\_2 は、2番目のパイプラインから作成されたマッピングを指します。

## PowerCenter からのコマンドタスクのインポート

コマンドタスクを PowerCenter からモデルリポジトリにインポートできます。

コマンドタスクをインポートする場合は、次の情報を考慮してください。

**PowerCenter のコマンドタスク内には複数のコマンドを追加できます。Developer tool には、コマンドタスクのすべてのコマンドを追加するためのフィールドが1つあります。**

PowerCenter では、コマンドタスク内に複数のコマンドを追加できます。Developer tool では、1つのフィールドにすべてのコマンドが含まれます。

**コマンドタスクをインポートする前に、[いずれかのコマンドが失敗した場合にタスクが失敗する] セッションプロパティをオンまたはオフにしたかを確認します。**

PowerCenter からモデルリポジトリにコマンドタスクをインポートする前に、[いずれかのコマンドが失敗した場合にタスクが失敗する] セッションプロパティをオンにしたかどうかを確認します。セッション

プロパティに基づき、インポートプロセスで2つのアンパサンド（&&）または1つのセミコロン（;）が各コマンドの最後に付加される場合があります。ただし最後のコマンドを除きます。

コマンドタスクの【いずれかのコマンドが失敗した場合にタスクが失敗する】セッションプロパティを有効にした場合、インポートプロセスで、Developer tool の各コマンドにセミコロンが付加されます。ただし最後のコマンドを除きます。例: <1 番目のコマンド>; <2 番目のコマンド>

コマンドタスクの【いずれかのコマンドが失敗した場合にタスクが失敗する】セッションプロパティを有効にした場合、インポートプロセスで、Developer tool の各コマンドに2つのアンパサンド（&&）が付加されます。例: <1 番目のコマンド>&& <2 番目のコマンド>

## コマンドタスクのインポートに伴う変換動作

コマンド内でセミコロンまたはバックスラッシュを使用するときは、コマンドタスクのインポート時の変換動作を考慮してください。

次のシナリオは、コマンドタスクを使用した変換動作を示しています。

- PowerCenter に次の2つのコマンドがあります。1 番目のコマンドは<C1;C2>のように最後にセミコロンがあり、2 番目のコマンドは<C3>です。

インポートプロセスでは、次の構文を使用してコマンドが変換されます。

“<C1;C2>;” “<C3>”

または

“<C1;C2>&&” “<C3>”

- PowerCenter に次の2つのコマンドがあります。1 番目のコマンドは<C1;C2\;>のように最後にバックスラッシュとセミコロンがあり、2 番目のコマンドは<C3>です。

インポートプロセスでは、次の構文を使用してコマンドが変換されます。

“<C1;C2\;>;” “<C3>”

または

“<C1;C2\;>&&” “<C3>”

- PowerCenter には次の2つのコマンドがあります。1 番目のコマンドは“<C1;C2\;>;”のように最後にバックスラッシュと2つのセミコロンがあり、2 番目のコマンドは“<C3>”です。

インポートプロセスでは、次の構文を使用してコマンドが変換されます。

“<C1;C2\;>;” “<C3>”または“<C1;C2\;>&&” “<C3>”

または

“<C1;C2\;>&&” “<C3>”

## ワークフローの詳細プロパティ

ワークフローの詳細プロパティには、ワークフローインスタンスの実行方法を定義するプロパティが含まれています。

### トレースレベル

ワークフローログに表示される詳細情報の量を決定します。トレースレベルの値は選択が可能です。また、トレースレベルをパラメータに割り当て、ワークフローパラメータでプロパティの値を定義することもできます。トレースレベルのデータ型は string です。

デフォルトは [INFO] です。

以下の表に、ワークフローのトレースレベルを示します。

トレースレベル	説明
エラー (ERROR)	ワークフローインスタンスが失敗する原因となったエラーメッセージを記録します。 ワークフローログでは SEVERE と表示されます。
注意 (WARNING)	エラーレベルのメッセージに加えて、警告メッセージも記録します。警告メッセージは、エラーが発生したが、ワークフローインスタンスの失敗の原因とはなかったことを示します。 ワークフローログでは、このレベルは WARNING と表示されます。
情報 (INFO)	警告レベルのメッセージに加えて、初期化情報とワークフローインスタンスの実行に関する詳細を記録します。タスクに渡された入力データ、タスクで実行された作業項目、タスクで生成された出力データなど、タスクの処理に関する詳細が記録されます。また、パラメータファイルの名前、および条件付きシーケンスフローの式の評価結果も記録されます。 ワークフローログでは、このレベルは INFO と表示されます。
トレース (TRACE)	情報レベルのメッセージに加えて、ワークフローやタスクの初期化に関する詳細を記録します。 ワークフローログでは、このレベルは FINE と表示されます。
デバッグ (DEBUG)	トレースレベルのメッセージに加えて、タスク入力やタスク出力の詳細とワークフローの状態に関する追加の詳細を記録します。 ワークフローログでは、このレベルは FINEST と表示されます。

### リカバリの有効化

ワークフローでリカバリが有効になっていることを示します。ワークフローのリカバリを有効にすると、再開リカバリストラテジが設定されたタスクでリカバリ可能なエラーが発生した場合、ワークフローインスタンスをキャンセルした場合、またはデータ統合サービスプロセスが予期せずにシャットダウンした場合に、ワークフローインスタンスをリカバリできます。ワークフローのリカバリを有効にする場合、ワークフローのタスクごとにリカバリストラテジを定義する必要があります。

デフォルトでは無効になっています。

### ワークフローの自動リカバリ

データ統合サービスプロセスが、予期せぬサービスプロセスのシャットダウンで中断されたワークフローインスタンスを自動的にリカバリすることを示します。ワークフローリカバリは、データ統合サービスプロセスの再開後に開始されます。ワークフローのリカバリが有効になっている場合、このオプションを選択できます。

デフォルトでは無効になっています。

# ワークフローのデプロイメント

Developer ツールでワークフローを開発するときは、ワークフロー定義を作成します。ワークフローのインスタンスを実行するには、アプリケーションにワークフロー定義を追加し、アプリケーションを Data Integration Service にデプロイします。

ユーザーが `infacmd wfs startWorkflow` コマンドを使用してワークフローを実行できるようにするには、ワークフローをデプロイします。ワークフローをデプロイすると、そのワークフロー専用のランタイムメタデータのセットがモデルリポジトリに作成されます。Developer tool でデプロイ後のワークフロー定義に変更を加える場合、変更を有効にするには、ワークフロー定義が含まれているアプリケーションを再デプロイするか、ワークフローを更新するアプリケーションをデプロイする必要があります。

ワークフローのデプロイは Developer ツールで行います。ワークフローをデプロイする手順は、モデルリポジトリの他のオブジェクトをデプロイする場合と同じです。

## ワークフローのデプロイと実行

ワークフローを Data Integration Service にデプロイすると、ワークフローの 1 つのインスタンスをデプロイ後すぐに実行できます。ワークフローをデプロイして実行するときにパラメータファイルを指定することはできません。ワークフローでパラメータを使用する場合は、デフォルトのパラメータ値が使用されます。

デプロイ後すぐにワークフローを実行するには、**【オブジェクトを実行】** ダイアログで **【デプロイ完了】** をクリックします。デプロイ済みのアプリケーションにワークフローが複数含まれている場合は、実行するワークフローを選択します。選択した各ワークフローのインスタンスが同時に実行されます。デプロイ済みのアプリケーションに他のタイプのオブジェクトが含まれている場合、それらのオブジェクトを実行対象として選択することはできません。

ワークフローインスタンスの実行は管理者ツールの **【監視】** タブで監視します。ワークフローの他のインスタンスを実行するには、`infacmd wfs startWorkflow` コマンドを使用します。

ワークフローをデプロイして実行したときにエラーメッセージが表示された場合は、ワークフローログと Data Integration Service のログで詳細を確認してください。

## ワークフローの実行

ワークフローをデプロイしたら、`infacmd wfs startWorkflow` コマンドを使用して、デプロイ済みのアプリケーションからワークフローのインスタンスを実行します。ワークフローの実行に使用するパラメータファイルを指定することができます。

デプロイ済みのアプリケーションから同じワークフローの複数のインスタンスを同時に実行できます。ワークフローインスタンスを実行すると、アプリケーションから Data Integration Service に要求が送信されます。Data Integration Service で、オブジェクトを接続するシーケンスフローに従ってワークフローのオブジェクトが実行されます。

例えば、次のコマンドでは、デプロイ済みアプリケーション `MyApplication` のワークフロー `MyWorkflow` のインスタンスが、パラメータファイル `MyParameterFile` で定義されたパラメータの値を使用して実行されます。

```
infacmd wfs startWorkflow -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a
MyApplication -wf MyWorkflow -pf MyParameterFile.xml
```



## ワークフローの監視

ワークフローインスタンスの実行は Monitoring ツールで監視します。Monitoring ツールは、管理者ツールの **【モニタリング】** タブへ直接リンクしています。

Monitoring ツールでは、実行中のワークフローとワークフローオブジェクトインスタンスのステータスが表示されます。Monitoring ツールでワークフローインスタンスの実行を強制終了したりキャンセルしたりできます。また、Monitoring ツールを使用して、ワークフローインスタンスのログを表示したりワークフローレポートを表示したりできます。

## ワークフローの削除

不要になったワークフローは削除することができます。ワークフローを削除すると、そのワークフロー内のすべてのオブジェクトが削除されます。

Developer ツールでワークフローを削除するときは、モデルリポジトリのワークフロー定義を削除します。ワークフロー定義が Data Integration Service にデプロイされていれば、デプロイ済みのワークフロー定義からワークフローのインスタンスの実行を継続できます。

ワークフローを削除するには、**Object Explorer** ビューでワークフローを選択してから、**【編集】** > **【削除】** の順にクリックします。

## ワークフローの例

ワークフローを開発する例を次に示します。

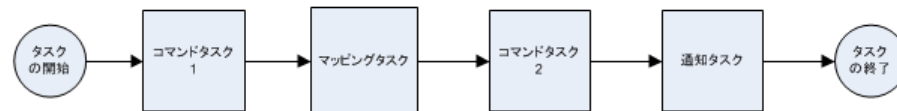
### 例: マッピングの実行前と実行後のコマンドの実行

マッピングの実行前と実行後にコマンドを使用してステップを実行するワークフローを開発することができます。例えば、マッピングタスクの前にコマンドタスクを設定して、マッピングが実行される前にマッピングタ



ターゲットに対するインデックスをドロップできます。マッピングタスクの後にコマンドタスクを設定して、マッピングの完了時にインデックスを再作成できます。

次の図のワークフローでは、コマンドを実行し、マッピングを実行し、別のコマンドを実行し、ワークフローのステータスをユーザーに通知する電子メールを送信します。



パラメータファイルを使用すると、ワークフローを実行するたびに柔軟にパラメータ値を変更することができます。このワークフローでは次のパラメータを使用できます。

- 最初のコマンドタスクが実行するコマンドを表すワークフローパラメータ。
- マッピングのソースへの接続を表すマッピングパラメータ。
- マッピングのターゲットへの接続を表すマッピングパラメータ。
- 2 番目のコマンドタスクが実行するコマンドを表すワークフローパラメータ。
- 通知タスクで電子メールを送信する電子メールアドレスを表すワークフローパラメータ。

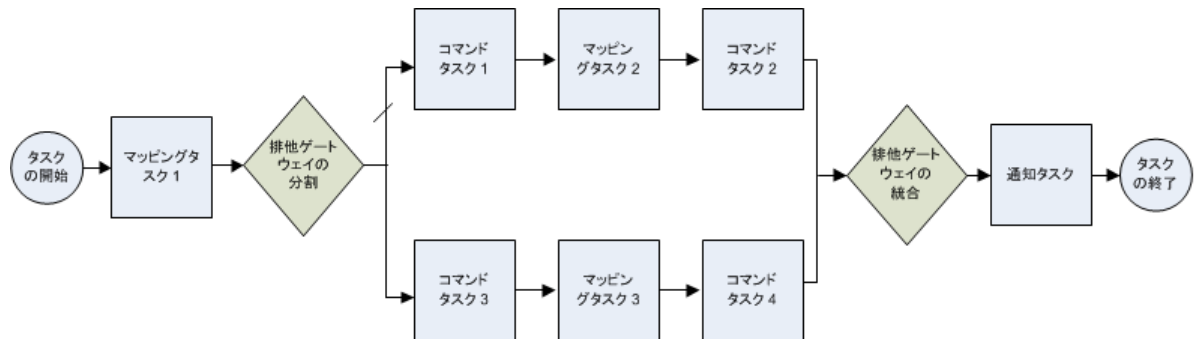
パラメータファイルでパラメータの値を定義します。ワークフローを実行するときにパラメータファイルを指定します。他のコマンドを実行する、他のソースまたはターゲットにマッピングを接続する、あるいは他のユーザーに電子メールを送信するには、それぞれ別のパラメータファイルを指定してワークフローを実行します。

## 例: 複数のシーケンスフローの作成

1 つのシーケンスフローを複数のシーケンスフローに分割し、ワークフローデータのパスを決定する条件を使用するワークフローを開発できます。ゲートウェイを使用してシーケンスフローを作成します。例えば、マッ

ピングが正常に実行された場合に、あるシーケンスフローに従い、マッピングが失敗した場合は別のシーケンスフローに従うワークフローを開発できます。

次の図に、排他ゲートウェイを使用してシーケンスフローを作成するワークフローを示します。



ワークフローには、次のコンポーネントが含まれます。

- 開始イベントと終了イベント。
- マッピングタスク。タスクはマッピングを実行し、「成功」出力をワークフローのブール変数に割り当てます。
- 2つの出力シーケンスフローを指定する排他ゲートウェイ。シーケンスフローの一方には、ワークフロー変数の値を評価する条件が含まれます。条件が true と評価されると、データ統合サービスはこのシーケンスフローの次のタスクを実行します。条件が false と評価されると、もう一方のシーケンスフローの次のタスクが実行されます。

この例では、各シーケンスフローにコマンドタスク、マッピングタスク、およびもう 1 つのコマンドタスクが含まれます。

- シーケンスフローを 1 つのフローにマージする排他ゲートウェイ。
- ワークフローのステータスをユーザーに通知する電子メールを送信する通知タスク。

## 第 2 章

# ワークフロー変数

この章では、以下の項目について説明します。

- [ワークフロー変数の概要, 27 ページ](#)
- [タスク入力, 28 ページ](#)
- [タスク出力, 28 ページ](#)
- [システムワークフロー変数, 29 ページ](#)
- [ユーザー定義ワークフロー変数, 29 ページ](#)
- [ワークフロー変数の使用場所, 32 ページ](#)
- [ワークフロー変数のデータ型変換, 36 ページ](#)

## ワークフロー変数の概要

ワークフロー変数は、ワークフローの実行時に変更できる値を表す変数で、値やレコードのランタイム情報を参照するために使用します。

ワークフロー変数には、システム変数とユーザー定義変数があります。システムワークフロー変数は、ワークフローインスタンス ID、ワークフローを開始したユーザー、ワークフローの開始時刻など、システムのランタイム情報を返します。

ユーザー定義ワークフロー変数では、タスク出力を取得したり、指定した条件を取得したりできます。ユーザー定義ワークフロー変数を作成した後に、変数にランタイム値を割り当てるようにワークフローを設定します。

タスクとワークフローの間でデータをやり取りするには、ワークフロー変数をタスク入力およびタスク出力に割り当てます。

ワークフロー変数は次の目的で使用できます。

**次のオブジェクトを実行するかどうかをランタイムデータに基づいて決める。**

変数の値を評価して次に実行するオブジェクトを決める場合は、条件付きシーケンスフローの式でワークフロー変数を使用します。例えば、「成功」タスク出力を取得するためのユーザー定義のブール変数を作成します。ゲートウェイの条件付きシーケンスフローの式でその変数を使用して、前のタスクが正常に実行されたかどうかを評価します。正常に実行された場合はタスク A を、正常に実行されなかった場合はタスク B を実行します。

**タスクフィールドの値にランタイムデータを使用する。**

実行時にフィールドに変数の値を使用する場合は、タスクフィールドでワークフロー変数を使用します。例えば、ワークフローを開始したユーザーに電子メールを送信するには、通知タスクの受信者のリストで UserName システム変数を使用します。

## タスク入力

タスク入力は、ワークフローのパラメータや変数からタスクに渡されるデータです。タスクで作業単位を実行するときにこの入力データが使用されます。

タスクを設定するときに、タスクに必要なワークフローのパラメータや変数を指定します。これにより、タスクの開始時に、ワークフローのパラメータや変数の値がタスクにコピーされます。

一部のタスクでは、**【入力】** タブでタスクに必要なワークフローのパラメータや変数を指定できます。それ以外のタスクでは、他のタブでタスクに必要なワークフローのパラメータや変数を指定します。

### 関連項目：

- [「タスク入力へのワークフローパラメータの割り当て」](#)（ページ 40）
- [「タスク入力への変数の割り当て」](#)（ページ 33）

## タスク出力

マッピングタスク、コマンドタスク、通知タスク、またはヒューマンタスクを設定する場合は、タスク出力を定義することができます。タスク出力は、タスクからワークフロー変数に渡されるデータです。

タスクを設定するときに、ワークフロー変数に割り当てるタスクの出力値を指定します。これにより、タスクの完了時に、タスクの出力値がワークフロー変数にコピーされます。Data Integration Service で条件付きシーケンスフローの式を評価したり、ワークフローの他のオブジェクトを実行したりする際に、ワークフロー変数からそれらの値にアクセスすることができます。

例えば、各タスクにはタスクが正常に実行されたかどうかを示す「成功」出力値が含まれています。この出力データにワークフローから直接アクセスすることはできません。このデータをこれ以降のワークフローで使用するには、TaskSuccessful という名前のワークフローブール変数を作成して、「成功」出力をその変数に割り当てます。その TaskSuccessful ワークフロー変数を、条件付きシーケンスフローの式に使用します。これにより、前のタスクが正常に実行された場合にワークフローの次のオブジェクトが実行されるようになります。

タスクでは、全般出力とタスク固有の出力が生成されます。タスクが失敗した場合は、タスクの全般出力の値がワークフロー変数にコピーされ、タスク固有の出力値はコピーされません。タスクが強制終了された場合には、タスクの出力値はワークフロー変数にコピーされません。

次の表に、各タスクで生成される全般出力を示します。

出力データ	データ型	説明
開始時刻	Date	タスクの実行が開始された日時。
終了時刻	Date	タスクの実行が完了した日時。
成功	Boolean	タスクが正常に実行されたかを示します。

**注:** 割り当てタスクでは、全般出力やタスク固有の出力は生成されません。

### 関連項目：

- [「マッピングタスクの出力」](#)（ページ 90）
- [「タスク出力を使用した値の割り当て」](#)（ページ 31）
- [「コマンドタスクの出力」](#)（ページ 64）

- [「通知タスクの出力」](#) (ページ 117)

## システムワークフロー変数

システムワークフロー変数は、システムのランタイム情報を返します。

システムワークフロー変数を作成することはできません。ワークフローで使用できる一連のシステムワークフロー変数は、Developer ツールで事前に定義されています。

条件付きシーケンスフローまたはタスクフィールドでシステムワークフロー変数を使用すると、実行時に変数の値が使用されるようになります。例えば、ワークフローを実行するユーザーに電子メールを送信するには、通知タスクの受信者のリストで UserName システム変数を使用します。

次の表に、システムワークフロー変数を示します。

システム変数	データ型	説明
InstanceID	String	ワークフローインスタンスの一意の ID。
StartTime	Date	ワークフローインスタンスの実行を開始する日時。
UserName	String	ワークフローインスタンスを実行するユーザーの名前。

## ユーザー定義ワークフロー変数

タスク出力を取得したり、指定した条件に基づいてワークフローの判定を行ったりするには、ユーザー定義ワークフロー変数を作成します。ユーザー定義ワークフロー変数のデータ型は、論理値、日付、整数、または文字列にすることができます。

ユーザー定義ワークフロー変数を使用するには、以下の手順を実行します。

1. ワークフロー変数を作成して初期値を設定します。  
ワークフローを開始するときは変数の初期値が使用されます。
2. 変数にランタイム値を割り当てるようにワークフローを設定します。  
ワークフローが開始された後は、ワークフローの設定に応じて、変数の初期値を計算したり変更したりできます。ユーザー定義変数の値は、割り当てタスクを使用して割り当てることができます。また、タスク出力を使用して割り当てすることもできます。
3. 条件付きシーケンスフローまたはタスクフィールドで変数を使用します。実行時に変数の値が使用されるようになります。

## ユーザー定義変数の作成

ユーザー定義ワークフロー変数を作成してランタイム情報を記録します。

1. エディタでワークフローを開きます。
2. ワークフローのプロパティまたはタスクのプロパティでユーザー定義ワークフロー変数を作成します。

- ワークフローの【プロパティ】ビューで【変数】タブをクリックします。【ユーザー】ビューで【追加】をクリックします。
- タスクの【プロパティ】ビューで、【タブ】タブまたは【出力】タブを選択します。【値】または【変数】カラムで【新しい変数】を選択します。

【変数の追加】ダイアログボックスが表示されます。

- 変数の名前と、必要に応じて説明を入力します。
- 変数のデータ型を選択します。
- 変数の初期値を入力します。
- 【OK】をクリックします。

## 割り当てタスクを使用した値の割り当て

割り当てタスクは、ユーザー定義ワークフロー変数に値を割り当てるタスクです。

ユーザー定義ワークフロー変数を作成するときは、初期値を入力します。ワークフローを開始するときは変数の初期値が使用されます。ワークフローに割り当てタスクを追加して、変数に別の値を割り当てるができます。以降のワークフローで、変数に割り当て値が使用されるようになります。

例えば、カウンタ変数を作成して初期値を 0 に設定し、割り当てタスクで現在の値に 1 を加えるように設定すると、変数の値を 1 ずつ増分することができます。

ワークフローに割り当てタスクを追加するときは、値を変更するユーザー定義変数を選択し、選択した変数に値を割り当てる式を記述します。

次の表に、ユーザー定義変数に割り当てることができる値を示します。

値	例
リテラル値	例えば、値 500 をユーザー定義変数に割り当てるには、式で次の値を入力します。 500
ワークフローパラメータ	例えば、ワークフローパラメータの値をユーザー定義変数に割り当てるには、式で次の値を入力します。 \$par:MyParameter
システム定義またはユーザー定義のワークフロー変数	例えば、システム定義またはユーザー定義のワークフロー変数の値を別のユーザー定義変数に割り当てるには、式で次の値を入力します。 \$var:MyVariable
トランスフォーマーメソッド言語の関数や演算子を使用した任意の有効な式	式の戻り値は、論理値、日付、整数、または文字列でなければなりません。戻り値のデータ型がそれ以外のデータ型の場合は、変換関数を使用して、サポートされているいずれかのデータ型に変換します。 例えば、式の値をユーザー定義変数に割り当てるには、式で次の値を入力します。 LENGTH('test') 式で等号演算子 (=) を使用する場合は、Data Integration Service で式の両辺が等しいかどうかチェックされ、true または false が返されます。例えば、次の式では、選択したユーザー定義変数に true または false のいずれかが割り当てられます。 \$var.MyVariable = 7 + 5

システムワークフロー変数に値を割り当てることはできません。

## 割り当て式のパラメータと変数

ユーザー定義ワークフロー変数に割り当てる式の値にワークフローのパラメータや変数を含めることができます。

ワークフローのパラメータや変数は、**割り当て式エディタ**の**【入力】**タブで選択できます。また、所定の構文に従って、式の中にパラメータ名または変数名を入力することもできます。

例えば、Counter という名前のユーザー定義ワークフロー変数を作成して初期値を 0 に設定し、割り当てタスクを使用して変数の値を 1 ずつ増分するとします。この場合、割り当てタスクで次の式を入力します。

```
$var:Counter + 1
```

Data Integration Service は、割り当て式の文字列リテラルに含まれるワークフロー変数またはパラメータ変数を解決しません。例えば、割り当てタスクを使用して変数に次の値を割り当てるとします。

```
'The mapping completed successfully: ${var:MappingIsSuccessful}'
```

Data Integration Service は文字列リテラルを評価しません。また、MappingIsSuccessful 変数値も解決しません。Data Integration Service は文字列に変数名を表示します。

## 割り当てタスクの設定

割り当てタスクを使用してユーザー定義ワークフロー変数に値を割り当てるには、事前にワークフロー変数を作成して初期値を設定しておく必要があります。

1. ワークフローに割り当てタスクを追加します。
2. エディタで割り当てタスクを選択します。
3. **【プロパティ】** ビューで **【割り当て】** タブをクリックします。
4. **【ユーザー定義変数】** カラムで、ユーザー定義ワークフロー変数を選択します。
5. **【式】** カラムの矢印をクリックします。

**割り当て式エディタ**が表示されます。

6. 変数に割り当てる値または式を入力します。  
**【関数】** タブにトランスフォーメーション言語関数が表示され、**【入力】** タブにワークフローのパラメータと変数が表示されます。関数、パラメータ、または変数の名前をダブルクリックして式に含めます。必要に応じて、演算子およびリテラル値を式に入力します。
7. **【検証】** ボタンを使用して式を検証します。  
ダイアログボックスにエラーが表示されます。
8. エラーを修正し、式を再度検証します。
9. **【OK】** をクリックします。

## タスク出力を使用した値の割り当て

タスクで生成される出力データを以降のワークフローに渡す場合は、ユーザー定義ワークフロー変数にタスク出力を割り当てます。

ユーザー定義ワークフロー変数を作成するときは、初期値を入力します。ワークフローを開始するときは変数の初期値が使用されます。タスクの**【出力】**タブを使用して、変数に別の値を割り当てます。タスクが完了すると、Data Integration Service は以降のワークフローで、変数にタスクの出力値を使用ようになります。

例えば、CommandStdOutput という名前のワークフロー変数を作成し、初期値を "test" に設定するとします。さらに、コマンドタスクの**【出力】**タブで、CommandStdOutput ワークフロー変数をコマンドから返される標準の出力に割り当てます。この場合、ワークフローの開始時に、ワークフロー変数の値が "test" に設定されます。コマンドタスクで echo コマンドを使用して CommandStdOutput 変数の値を出力すると、Data

Integration Service は初期値の"test"を出力します。コマンドタスクが完了すると、Data Integration Service はコマンドから返される標準の出力にワークフロー変数の値を設定します。

システムワークフロー変数にタスク出力を割り当てることはできません。

## 関連項目：

- [「マッピングタスクの出力」 \(ページ 90\)](#)
- [「タスク出力」 \(ページ 28\)](#)
- [「コマンドタスクの出力」 \(ページ 64\)](#)
- [「通知タスクの出力」 \(ページ 117\)](#)

## タスク出力の割り当て

ユーザーが定義したワークフロー変数にタスクの出力値を割り当てることができます。

1. エディタでワークフローを開きます。
2. エディタで出力データを生成するタスクを選択します。
3. **【プロパティ】** ビューで **【出力】** タブをクリックします。  
タスクで生成されるすべての出力データがタブに一覧表示されます。
4. 出力を検索するための文字列を入力します。  
この文字列にはワイルドカード文字を使用できます。文字列の大文字と小文字は区別されません。
5. 出力の **【変数】** カラムをクリックします。
6. 変数名を選択するか、**【新しい変数】** をクリックして新しい変数を作成し、出力に割り当てます。
7. 出力の割り当てをクリアするには、出力を選択し、**【クリア】** をクリックします。または、**【すべてクリア】** をクリックして、すべての出力の割り当てをクリアします。

# ワークフロー変数の使用場所

データ統合サービスが変数の値を評価して次に実行するオブジェクトを決める場合は、条件付きシーケンスフローの式でワークフロー変数を使用します。データ統合サービスがフィールドに変数の値を使用する場合は、タスクフィールドでワークフロー変数を使用します。

式やタスクフィールドに応じて、ワークフロー変数名を選択できる場合と入力できる場合があります。

次の表に、ワークフロー変数を使用できるオブジェクトおよびフィールドを示します。

オブジェクト	タブまたはダイアログボックス	フィールド	選択または入力
シーケンスフロー	[条件] タブ	条件	両方
割り当てタスク	[割り当て式エディタ] ダイアログボックス	式	両方
コマンドタスク	[コマンド] タブ	コマンド	両方



オブジェクト	タブまたはダイアログボックス	フィールド	選択または入力
コマンドタスク	[入力] タブ	タスク入力に割り当てられた詳細設定プロパティ	選択
ヒューマンタスク	[入力] タブ	処理された項目数	選択
マッピングタスク	[入力] タブ	ユーザー定義のマッピングパラメータ タスク入力に割り当てられた詳細設定プロパティ	選択
通知タスク	[通知] タブ	動的な電子メールのコンテンツ	入力
通知タスク	[メールプロパティ] ダイアログボックス	動的な受信者 動的な電子メールアドレス	選択
通知タスク	[メールプロパティ] ダイアログボックス	動的な電子メールのコンテンツ	両方

## タスク入力への変数の割り当て

マッピングタスクとコマンドタスクでは、**[入力]** タブでタスクに必要なワークフロー変数を指定できます。

マッピングタスクおよびコマンドタスクの **[入力]** タブでは、タスクの設定プロパティをタスク入力に割り当てて、ワークフロー変数のプロパティの値を定義できます。タスクの設定プロパティは、タスクの **[詳細]** タブに一覧表示されます。

マッピングタスクの **[入力]** タブでは、ワークフロー変数にユーザー定義のマッピングパラメータを割り当てて、ユーザー定義のマッピングパラメータの値にワークフローのランタイムデータを使用することができます。

1. エディタでマッピングタスクまたはコマンドタスクを選択します。
2. **[プロパティ]** ビューで **[詳細]** タブをクリックして、詳細設定プロパティをタスク入力に割り当てます。  
プロパティの **[値]** カラムで、**[タスク入力に割り当て]** を選択します。
3. **[入力]** タブをクリックします。
4. 入力を検索するための文字列を入力します。  
この文字列にはワイルドカード文字を使用できます。文字列の大文字と小文字は区別されません。
5. 設定プロパティまたはマッピングパラメータの **[値]** カラムをクリックします。
6. 既存のワークフロー変数、新しいワークフロー変数、またはリテラル値にプロパティまたはパラメータを割り当てます。
  - ワークフロー変数の名前を選択します。
  - **[新しい変数]** をクリックします。 **[変数の追加]** ダイアログボックスで、ワークフロー変数の名前、タイプ、および初期値を入力します。ワークフロー変数が作成され、プロパティに割り当てられます。
  - **[新しい値]** をクリックします。 **[値の追加]** ダイアログボックスで、プロパティに割り当てるリテラル値とデータ型を入力します。
7. 入力の割り当てをクリアするには、入力を選択し、**[クリア]** をクリックします。または、**[すべてクリア]** をクリックして、すべての入力の割り当てをクリアします。

関連項目：

- [「マッピングタスクの入力」 \(ページ 86\)](#)
- [「タスク入力」 \(ページ 28\)](#)
- [「コマンドタスクの入力」 \(ページ 63\)](#)

## 式および文字列の変数名

式や文字列フィールドでワークフロー変数を使用する場合は、**【入力】** タブで名前を選択するか、所定の構文に従って名前を入力できます。

次の表に、式および文字列フィールドにワークフロー変数名を入力する際の構文を示します。

フィールド	構文	例
条件付きシーケンスフローまたは割り当てタスクの式	<ul style="list-style-type: none"><li>- <code>\$var:&lt;variable_name&gt;</code> (ユーザー定義変数)</li><li>- <code>\$var:sys.&lt;variable_name&gt;</code> (システム変数)</li></ul>	<p>例えば、<code>CommandExitCode</code> という名前のワークフロー変数を作成し、その変数にコマンドタスクの終了コードの出力値を割り当てるとします。コマンドタスクをマッピングタスクに接続する条件付きシーケンスフローで次の式を作成します。</p> <pre>\$var:CommandExitCode = 0</pre> <p>この場合、Data Integration Service で条件が評価され、前のコマンドタスクの終了コードでコマンドが正常に実行されたことを示す 0 が返された場合にマッピングタスクが実行されます。</p>
コマンドタスクまたは通知タスクの文字列フィールド	<ul style="list-style-type: none"><li>- <code>\${var:&lt;variable_name&gt;}</code> (ユーザー定義変数)</li><li>- <code>\${var:sys.&lt;variable_name&gt;}</code> (システム変数)</li></ul>	<p>コマンドタスクまたは通知タスクの文字列フィールドに変数名を入力するときは、変数名を中括弧で囲む必要があります。例えば、<code>MappingErrorRows</code> という名前のワークフロー変数を作成し、その変数にマッピングタスクのエラー行の数を示す出力値を割り当てるとします。通知タスクの本文に次のテキストを入力します。</p> <pre>Mapping failed to write \${var:MappingErrorRows} rows to the target.</pre>

変数名に"var:"を含めないと、名前がパラメータとして使用されます。例えば、`$CommandExitCode` または `${CommandExitCode}` と入力すると、`$par:CommandExitCode` または `${par:CommandExitCode}` が使用されます。

## 文字列内のエスケープ文字

文字列フィールドでワークフロー変数を使用する場合、エスケープ文字を使用すると、変数の値を解決せずに、ワークフロー変数名を文字列として表示することができます。

ワークフロー変数名の `${...}` という形式の構文の前に、エスケープ文字としてバックスラッシュ (`\`) を使用します。

例えば、`myVariable` という名前のワークフロー文字列変数の値が `"test"` であるとして、通知タスクの本文フィールドに次のテキストを入力します。

```
Variable \${var:myVariable} has a value of ${var:myVariable}
```

ワークフローを実行すると、電子メールの本文フィールドに次の文字列が表示されます。

```
Variable ${var:myVariable} has a value of test
```

## ディレクトリパス内のエスケープ文字

ディレクトリパス内でワークフロー変数名を使用する場合は、ディレクトリパスのバックスラッシュの前にエスケープ文字を使用します。

次の表に、ディレクトリパスでのエスケープ文字と変数名の使用例を示します。

文字列フィールドの構文	出力値	説明
C:\\${var:myVariable}	C:\\${var:myVariable}	変数名が文字列として表示されます。
C:\\\${var:myVariable}	C:\test	バックスラッシュが通常の文字として読み取られ、変数が値に解決されます。
C:\temp\\\${var:myVariable}	C:\temp\test	バックスラッシュが通常の文字として読み取られ、変数が値に解決されます。最初のバックスラッシュにはエスケープ文字は不要です。
C:\\\\${var:myVariable}	C:\\${var:myVariable}	バックスラッシュが通常の文字として読み取られ、変数名が文字列として表示されます。
C:\\\\\\${var:myVariable}	C:\\test	2つのバックスラッシュが通常の文字として読み取られ、変数が値に解決されます。

## コマンドタスク内のエスケープ文字

コマンド出力をファイルに書き込むには、コマンドタスクを使用します。コマンド出力に変数名が含まれる場合には、バックスラッシュをエスケープ文字として使用し、変数名の前に「\$」文字を追加します。

コマンドタスクが Windows 以外のオペレーティングシステムで実行されるように設定した場合は、3つのバックスラッシュを使用します。最初のバックスラッシュは2番目のバックスラッシュのエスケープ文字です。3番目のバックスラッシュは「\$」文字のエスケープ文字です。

例えば、コマンドタスクで次のコマンドを定義します。

```
echo \\${var:myVariable} = ${var: myVariable} > file.txt
```

ワークフロー実行時、変数に 10 という値がある場合、コマンドタスクは次の文字列を file.txt に書き込みます。

```
${var:Var} = 10
```

## ネストされた変数

Data Integration Service は、1レベルの変数値を解決します。Data Integration Service は別のワークフローパラメータまたは変数内でネストされている変数値を解決しません。

例えば、次のデータ型と初期値を持つワークフロー変数を作成するとします。

- データ型が integer で初期値が 4 の Variable1
- データ型が integer で初期値が 3 の Variable2
- データ型が文字列で初期値が \${var:Variable1} + \${var:Variable2} の Variable3

式またはタスクフィールドで Variable3 を使用すると、Data Integration Service はネストされた Variable1 と Variable2 を 7 に解決しません。代わりに Variable3 に次のような文字列値が使用されます。

```
${var:Variable1} + ${var:Variable2}
```

# ワークフロー変数のデータ型変換

ワークフロー変数のデータ型は、論理値、日付、整数、または文字列にすることができます。Data Integration Service でデータ型を変換できる場合は、いずれかのデータ型の変数を、データ型が異なるワークフロー変数、パラメータ、リテラル値、タスク入力、またはタスク出力に割り当てることができます。

次の表に、Data Integration Service で実行されるワークフロー変数のデータ型変換を示します。

変数のデータ型	String	Integer	Boolean	Date
String	はい	はい	はい	いいえ
Integer	はい	はい	はい	いいえ
Boolean	はい	はい	はい	いいえ
Date	はい	いいえ	いいえ	はい

文字列を整数に変換するには、文字列が数値でなければなりません。

文字列を論理値に変換するには、文字列が"true"または"false"でなければなりません。

Data Integration Service による整数から論理値への変換では、値が 0 の場合は false に変換され、0 以外の場合はすべて true に変換されます。

Data Integration Service による論理値から整数への変換では、値が false の場合は 0 に変換され、true の場合は 1 に変換されます。

ワークフローを実行すると、データが有効なデータ型に変換されます。例えば、StartTime システムワークフロー変数のデータ型は日付です。この変数を通知タスクの本文の文字列フィールドで 사용할 ことができます。ワークフローを実行すると、システムワークフロー変数に格納された日付が文字列に変換されます。

## 日付変数の形式の変更

Data Integration Service では、データ型が日付のワークフロー変数に対し、DAY MON DD HH24:MI:SS YYYY という形式を使用します。このデフォルトの日付変数形式は、割り当てタスクを使用して変更することができます。

割り当てタスクを使用して、値を日付から、特定の日付形式の文字列値に変換します。次に、変換された値を文字列のワークフロー変数に割り当てます。

1. データ型が文字列のユーザ定義ワークフロー変数を作成します。
2. ワークフローの中の、日付ワークフロー変数にランタイム値を割り当てるタスクの後に、割り当てタスクを追加します。
3. この割り当てタスクをワークフロー内の他のオブジェクトに接続します。
4. エディタで割り当てタスクを選択します。
5. **【プロパティ】** ビューで **【割り当て】** タブをクリックします。
6. **【ユーザー定義変数】** カラムで、文字列ワークフロー変数を選択します。
7. **【式】** カラムの矢印をクリックします。

**割り当て式エディタ**が表示されます。

8. 次の式を入力して、日付ワークフロー変数の値を特定の日付形式の文字列値に変換します。

`TO_CHAR(date_variable [, format])`

例えば、次の式を入力します。

```
TO_CHAR($var:MyDateVariable, 'MM/DD/YYYY HH24:MI:SS')
```

9. **【検証】** をクリックして、この式を検証します。  
ダイアログボックスにエラーが表示されます。
10. エラーを修正し、式を再度検証します。
11. **【OK】** をクリックします。
12. 式フィールドまたはタスクフィールドに、文字列ワークフロー変数を使用します。

## 第 3 章

# ワークフローパラメータ

この章では、以下の項目について説明します。

- [ワークフローパラメータの概要, 38 ページ](#)
- [パラメータを使用してワークフローを実行するプロセス, 39 ページ](#)
- [ワークフローパラメータの使用場所, 39 ページ](#)
- [ユーザー定義マッピングパラメータのワークフローパラメータの作成, 42 ページ](#)
- [デフォルトのパラメータ値, 43 ページ](#)
- [ワークフローパラメータのデータ型変換, 44 ページ](#)
- [パラメータセット, 44 ページ](#)
- [パラメータファイル, 48 ページ](#)

## ワークフローパラメータの概要

ワークフローパラメータは、ワークフローの実行前に定義する定数値です。ワークフローパラメータを使用してワークフローのタスクの値、またはユーザー定義マッピングパラメータの設定をします。ワークフローパラメータを使用して、接続パラメータの値、または設定プロパティ、コマンド文字列、電子メールアドレスなどの文字列パラメータの値の設定もできます。

ワークフローのパラメータを作成すると、異なるパラメータ値でワークフローを実行できます。ワークフローの特定の属性を変更する必要がある場合、パラメータを使用すると複数のワークフローを作成するオーバーヘッドを削減できます。すべてのワークフローパラメータはユーザー定義パラメータです。

ワークフローパラメータをタスク入力に割り当てて、ワークフローからタスクにデータを渡すことができます。例えば、作業ディレクトリパラメータを定義して、そのパラメータをワークフローのコマンドタスクに割り当てます。

ワークフローパラメータは、マッピングタスクのユーザー定義マッピングパラメータに割り当てすることもできます。例えば、通知タスクで通知電子メールを送信するための電子メールアドレスを識別するワークフローパラメータを定義します。そのワークフローパラメータを通知タスクの受信者フィールドで参照します。

パラメータセットまたはパラメータファイルにパラメータを含めると、ワークフローパラメータの値をオーバーライドできます。パラメータセットはパラメータ値を含むリポジトリオブジェクトです。パラメータセットを使用してワークフローをデプロイできます。パラメータファイルはパラメータ値を含む XML ファイルです。パラメータファイルは、リポジトリではなく、ファイルシステムに存在します。ワークフローを実行する場合、ワークフロー実行用の特定のパラメータセットまたはパラメータファイルを指定できます。

## タスク入力

タスク入力は、ワークフローのパラメータや変数からタスクに渡されるデータです。タスクで作業単位を実行するときにこの入力データが使用されます。

タスクを設定するときに、タスクに必要なワークフローのパラメータや変数を指定します。これにより、タスクの開始時に、ワークフローのパラメータや変数の値がタスクにコピーされます。

マッピングタスク、コマンドタスク、およびヒューマンタスクには、タスクに必要なワークフローパラメータおよび変数を設定する【入力】ビューが含まれます。【入力】ビューで、マッピングタスクの設定プロパティのパラメータを設定できます。

異なるタスクビューで他のタスクのワークフローパラメータを参照できます。例えば、通知タスクの【通知】ビューで電子メールアドレスを含むワークフローパラメータを設定します。

出力ゲートウェイの条件付きシーケンスフローの式にワークフローパラメータを使用できます。データ統合サービスはパラメータ値を評価し、ワークフローで次に実行するオブジェクトを識別します。

## パラメータを使用してワークフローを実行するプロセス

ワークフローパラメータは、ワークフローの実行前に定義する定数値です。パラメータセットまたはパラメータファイルにパラメータを追加する場合、パラメータ値をオーバーライドできます。

異なるパラメータ値でワークフローを実行するには、次の手順を実行します。

1. ワークフローパラメータを作成し、デフォルト値を割り当てます。
2. タスク入力にパラメータを割り当てるか、マッピングパラメータにパラメータを割り当てます。
3. ワークフローで、ワークフローおよびマッピングパラメータを含む 1 つ以上のパラメータセットを作成します。必要に応じてパラメータ値を変更します。
4. ワークフローおよびパラメータセットをデータ統合サービスにデプロイします。
5. コマンドラインからワークフローを実行し、ワークフロー実行に使用するパラメータセットを指定します。

**注:** パラメータファイルを作成し、パラメータファイルを指定してコマンドラインからワークフローを実行できます。パラメータファイルとパラメータセットを同時に指定してワークフローを実行することはできません。

## ワークフローパラメータの使用場所

データ統合サービスがパラメータの値を評価して次に実行するオブジェクトを決める場合は、条件付きシーケンスフローの式でワークフローパラメータを使用します。データ統合サービスがフィールドにパラメータの値を使用する場合は、オブジェクトフィールドでワークフローパラメータを使用します。

式やタスクフィールドに応じて、ワークフローパラメータ名は選択するか、または直接入力します。

次の表に、ワークフローパラメータを使用できるオブジェクトおよびフィールドを示します。

オブジェクト	タブまたはダイアログボックス	フィールド	選択または入力
ワークフロー	[詳細] タブ	トレースレベル	選択
シーケンスフロー	[条件] タブ	条件	両方
割り当てタスク	[割り当て式エディタ] ダイアログボックス	式	両方
コマンドタスク	[コマンド] タブ	コマンド	両方
コマンドタスク	[入力] タブ	タスク入力に割り当てられた詳細設定プロパティ	選択
ヒューマンタスク	[入力] タブ	処理された項目数	選択
マッピングタスク	[入力] タブ	ユーザー定義のマッピングパラメータ タスク入力に割り当てられた詳細設定プロパティ	選択
通知タスク	[通知] タブ	動的な電子メールのコンテンツ	入力
通知タスク	[メールプロパティ] ダイアログボックス	動的な受信者 動的な電子メールアドレス	選択
通知タスク	[メールプロパティ] ダイアログボックス	動的な電子メールのコンテンツ	両方

## タスク入力へのワークフローパラメータの割り当て

マッピングタスク、ヒューマンタスク、およびコマンドタスクでは、**[入力]** タブでタスクに必要なワークフローパラメータを指定できます。

マッピングまたはコマンドタスクの **[入力]** タブで、タスクの設定プロパティをタスク入力に割り当ててワークフローパラメータのプロパティ値を定義できます。タスクの設定プロパティは、タスクの **[詳細]** タブに一覧表示されます。

マッピングタスクの **[入力]** タブでは、一部のユーザー定義のマッピングパラメータをワークフローパラメータに割り当てることもできます。ワークフローに複数回出現するユーザー定義のマッピングパラメータに、個別の値を割り当てることができます。

**注:** ワークフローパラメータのタイプは、接続タイプまたは文字列タイプである必要があります。

1. エディタでマッピング、コマンド、またはヒューマンタスクを選択します。
2. **[プロパティ]** ビューをクリックします。
3. マッピングタスクまたはコマンドタスクの場合、**[詳細]** タブをクリックして、詳細設定のプロパティをタスク入力に割り当てます。  
プロパティの **[値]** カラムで、**[タスク入力に割り当て]** を選択します。
4. **[入力]** タブをクリックします。
5. **[入力]** タブの上部で、更新するプロパティを検索します。

この検索文字列にはワイルドカード文字を使用できます。文字列の大文字と小文字は区別されません。



6. プロパティの【値】列をクリックします。
7. 【値】列で、既存のワークフローパラメータ、新規ワークフローパラメータ、またはリテラル値にプロパティを割り当てることを選択します。
  - 既存のワークフローパラメータ名を選択します。
  - ワークフローパラメータを作成します。【新しいパラメータ】をクリックします。【パラメータの追加】ダイアログボックスで、ワークフローパラメータの名前、タイプ、およびデフォルト値を入力します。Developer tool でワークフローパラメータが作成され、プロパティに割り当てられます。
  - 【新しい値】をクリックします。【値の追加】ダイアログボックスで、プロパティに割り当てるリテラル値とデータ型を入力します。
8. 入力の割り当てをクリアするには、入力を選択し、【クリア】をクリックします。または、【すべてクリア】をクリックして、すべての入力の割り当てをクリアします。

#### 関連項目：

- [「マッピングタスクの入力」 \(ページ 86\)](#)
- [「タスク入力」 \(ページ 28\)](#)
- [「コマンドタスクの入力」 \(ページ 63\)](#)

## 式および文字列のパラメータ名

式や文字列フィールドでワークフローパラメータを使用する場合は、【入力】タブで名前を選択するか、所定の構文に従って名前を入力できます。

次の表に、式および文字列フィールドにワークフローパラメータ名を入力する際の構文を示します。

フィールド	構文	例
条件付きシーケンスフローまたは割り当てタスクの式	<code>\$par:&lt;parameter_name&gt;</code>	例えば、条件付きシーケンスフローで次の式を作成するとします。 <code>\$par:Connection=SourceConnection</code> この場合、Data Integration Service で条件が評価され、パラメータの値が <code>SourceConnection</code> の場合に接続されたタスクが実行されます。
コマンドタスクまたは通知タスクの文字列フィールド	<code>\${par:&lt;parameter_name&gt;}</code>	コマンドタスクまたは通知タスクの文字列フィールドにパラメータ名を入力するときは、パラメータ名を中括弧で囲む必要があります。例えば、コマンドタスクの次のコマンドでは、 <code>SourceDirectory</code> という名前のワークフローパラメータを使用して、コピーするファイルのソースディレクトリを定義しています。 <code>copy \${par:SourceDirectory} H:\marketing\</code>

パラメータ名に "par:" を含まないと、名前がパラメータとして使用されます。例えば、`$SourceDirectory` または `${SourceDirectory}` と入力すると、`$par:SourceDirectory` または `${par:SourceDirectory}` が使用されます。

## 文字列のエスケープ文字

文字列フィールドでワークフローパラメータ名を使用する場合、エスケープ文字を使用すると、パラメータの値を解決せずに、ワークフローパラメータ名を文字列として表示することができます。

ワークフローパラメータ名の `${...}` という形式の構文の前に、エスケープ文字としてバックスラッシュ (`\\uff09`) を使用します。

例えば、myParameter という名前のワークフロー文字列パラメータの値が"test"であるとします。通知タスクの本文フィールドに次のテキストを入力します。

Parameter \\${par:myParameter} has a value of \${par:myParameter}

ワークフローを実行すると、電子メールの本文フィールドに次の文字列が表示されます。

Parameter \${par:myParameter} has a value of test

ディレクトリパス内でワークフローパラメータ名を使用する場合は、ディレクトリパスのバックスラッシュの前にエスケープ文字を使用します。

次の表に、ディレクトリパスでのエスケープ文字とパラメータ名の使用例を示します。

文字列フィールドの構文	出力値	説明
C:\\${par:myParameter}	C:\\${par:myParameter}	パラメータ名が文字列として表示されます。
C:\\\${par:myParameter}	C:\test	バックスラッシュが通常の文字として読み取られ、パラメータが値に解決されます。
C:\temp\\\${par:myParameter}	C:\temp\test	バックスラッシュが通常の文字として読み取られ、パラメータが値に解決されます。最初のバックスラッシュにはエスケープ文字は不要です。
C:\\\\\${par:myParameter}	C:\\${par:myParameter}	バックスラッシュが通常の文字として読み取られ、パラメータ名が文字列として表示されます。
C:\\\\\\\${par:myParameter}	C:\\test	2つのバックスラッシュが通常の文字として読み取られ、変数が値に解決されます。

## ネストされたパラメータ

Data Integration Service は、1 レベルのパラメータ値を解決します。Data Integration Service は別のワークフローパラメータまたは変数内でネストされているパラメータ値を解決しません。

例えばパラメータファイルで、ワークフローパラメータに次のように値を割り当てるとします。

- Parameter1 の値は 3
- Parameter2 の値は 4
- Parameter3 の値は\${par:Parameter1} + \${par:Parameter2}

式またはタスクフィールドで Parameter3 を使用すると、Data Integration Service はネストされた Parameter1 と Parameter2 を 7 に解決しません。代わりに Parameter3 に次のような文字列値が使用されます。

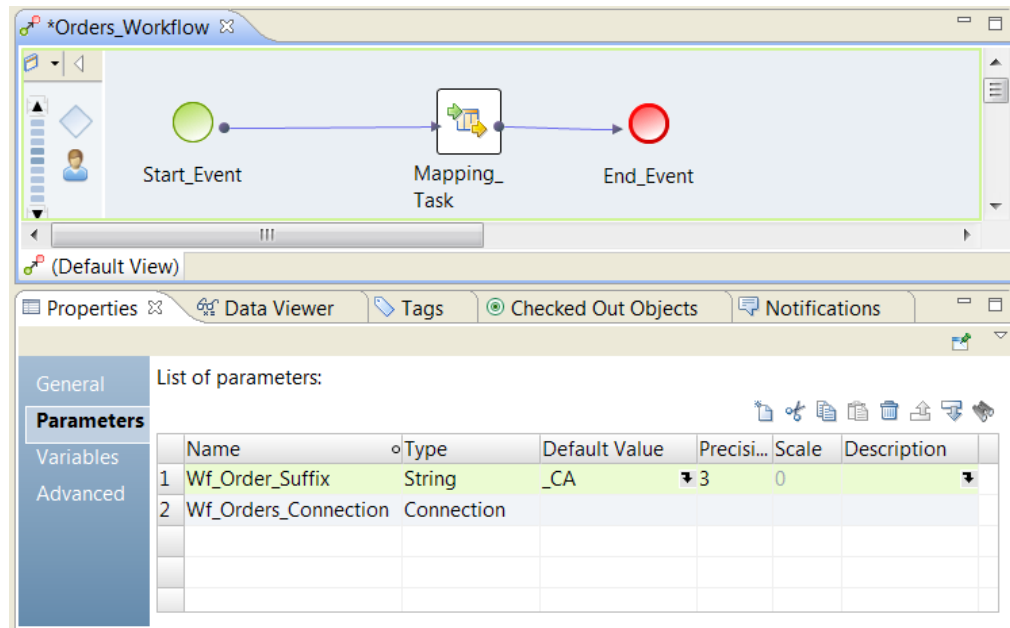
\${par:Parameter1} + \${par:Parameter2}

## ユーザー定義マッピングパラメータのワークフローパラメータの作成

ワークフローパラメータを作成してユーザー定義マッピングパラメータをオーバーライドできます。

1. エディタでワークフローを開きます。

- ワークフローの【プロパティ】ビューで【パラメータ】タブをクリックします。



- パラメータを追加するには、【新規】をクリックします。  
Developer tool により、デフォルトのプロパティを設定したパラメータが作成されます。必要に応じてパラメータプロパティの各フィールドを変更します。
- パラメータ名を入力します。
- 接続または文字列パラメータタイプを選択します。
- パラメータのデフォルト値を入力します。  
接続パラメータの場合、接続を選択します。文字列パラメータの場合、文字列値を入力します。
- 必要に応じてパラメータのスケールと説明を入力します。

## デフォルトのパラメータ値

ワークフローパラメータを作成するときは、デフォルト値を入力する必要があります。

パラメータセットまたはパラメータファイルを設定したワークフローを実行する場合、データ統合サービスはすべてのパラメータをパラメータセットまたはパラメータファイルの値セットに解決します。

データ統合サービスは、次の状況でパラメータをデフォルト値に解決します。

- パラメータセットもパラメータファイルも使用せずにワークフローを実行する場合。
- パラメータセットにもパラメータファイルにもパラメータ値を定義しない場合。

## ワークフローパラメータのデータ型変換

ワークフローパラメータには、接続パラメータと文字列パラメータの2種類があります。データ統合サービスによるデータ型変換が可能な場合、文字列ワークフローパラメータを integer または boolean 型のワークフロー変数やタスク入力に割り当てることができます。

次の表に、データ統合サービスで実行されるワークフローパラメータのデータ型変換を示します。

パラメータのデータ型	Integer	Boolean	Date
String	はい	はい	いいえ

文字列を整数に変換するには、文字列が数値でなければなりません。

文字列をブールに変換するには、文字列が「true」または「false」でなければなりません。

例えば、マッピングタスクに boolean データ型の高精度プロパティがあります。ワークフローパラメータをプロパティに割り当てる必要があります。デフォルト値「true」または「false」の文字列としてワークフローパラメータを定義できます。ワークフローを実行すると、データ統合サービスはパラメータ値をブール値に変換します。

接続パラメータを接続パラメータではないパラメータに割り当ててはできません。

## パラメータセット

パラメータセットはモデルリポジトリ内のオブジェクトであり、マッピングおよびワークフローの実行に使用する一連のパラメータとその値を含みます。

パラメータセットを作成する場合、パラメータを使用するマッピングまたはワークフローを選択します。マッピングまたはワークフローを選択後、パラメータセットのパラメータを手動で入力するか、マッピングまたはワークフローのリポジトリにすでにあるパラメータを選択できます。

さまざまな状況でパラメータセットを使用できます。例えば、テスト環境でワークフローを実行する場合に特定のパラメータセットを使用することがあります。

マッピング、マッピングタスク、またはワークフローでパラメータセットを使用します。アプリケーションをデプロイする場合、アプリケーションに1つ以上のパラメータセットを追加できます。複数のアプリケーションに1つのパラメータセットを追加し、デプロイすることができます。ワークフローまたはマッピングでパラメータセットを使用するには、ワークフローまたはマッピングのデプロイ時にアプリケーションにパラメータセットを追加する必要があります。

次の画像は、2つのマッピングのパラメータを含むパラメータセットを示します。

<

パラメータセットには次の情報が含まれます。

### オブジェクト名

パラメータの定義を含むマッピング、マップレット、またはワークフローの名前

### パラメータ名

マッピング、マップレット、またはワークフローのパラメータ名

### 値

ランタイムに使用するパラメータ値パラメータセットのパラメータ値は、マッピングまたはワークフローのパラメータ値をオーバーライドします。

### タイプ

パラメータのタイプ。パラメータタイプの例には、文字列、数値タイプ、接続、ポートリスト、ソートリスト、および日付/時刻パラメータが含まれます。

**注:** パラメータセットで指定したパラメータタイプが、マッピング、マッピングタスク、またはワークフロー内のパラメータタイプと一致している必要があります。パラメータタイプが一致しない場合、マッピング、マッピングタスク、またはワークフローはパラメータのデフォルト値を使用します。

パラメータセットを使用してマッピングパラメータ値を設定する場合、パラメータセットとマッピング間のリンクはマッピングが定義されたプロジェクトによって異なります。プロジェクト名が変更された場合は、リンクを再確立する必要があります。

リンクを再確立するには、パラメータセットを編集し、パラメータセットを使用するマッピングを再選択します。

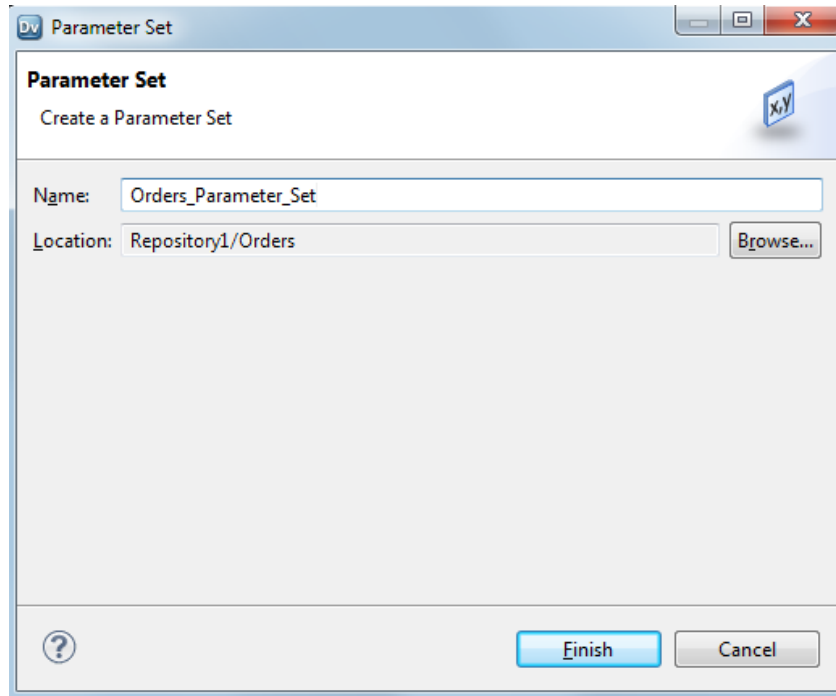
例えば、パラメータセット ps\_1 を使用するマッピング m\_1 を含む、P1 という名前のプロジェクトがあるとします。プロジェクト名を P2 に変更しました。その後、パラメータセットを編集し、マッピング m\_1 を再選択する必要があります。

## パラメータセットの作成

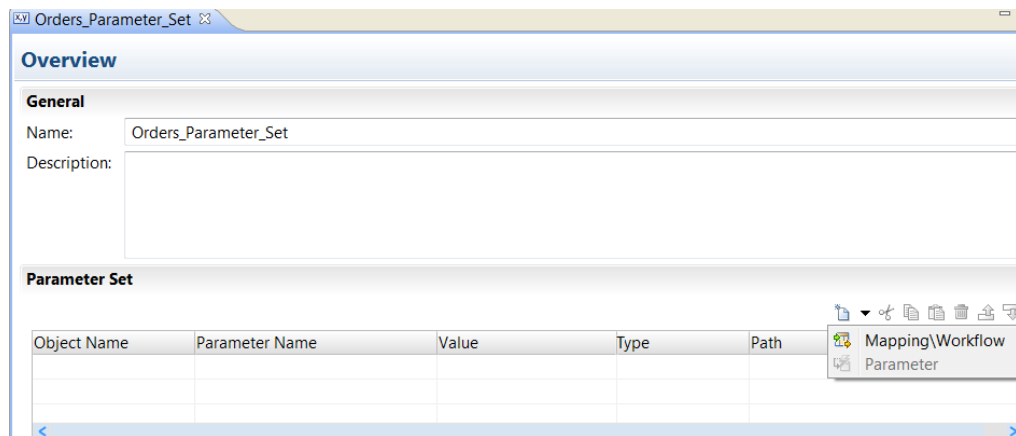
マッピングおよびワークフローのランタイムコンテキスト変更に使用できるパラメータセットを作成します。

パラメータセットを作成する場合、パラメータを含めるためのマッピングまたはワークフローを選択します。マッピングまたはワークフローを選択後、パラメータセットに手でパラメータを入力するか、パラメータを選択できます。

1. [オブジェクトエクスプローラ] ビューで、プロジェクトを右クリックし、[新規] > [パラメータセット] をクリックします。
2. パラメータセット名を入力して、[完了] をクリックします。

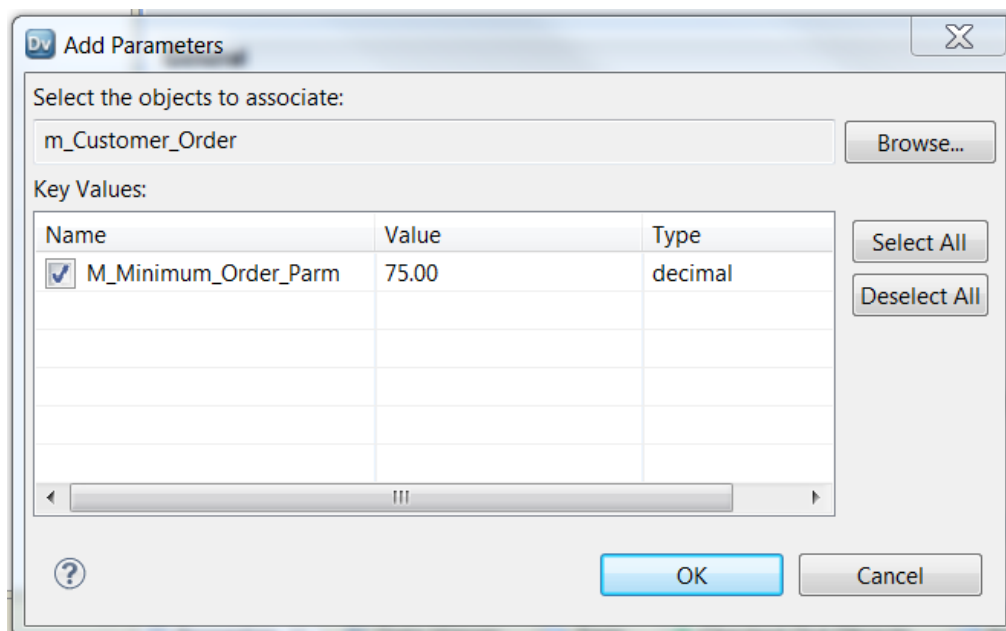


3. [プロパティ] パネルをドラッグしてグリッドを表示し、パラメータをパラメータセットに追加します。
4. [新規] > [マッピング/ワークフロー] をクリックします。

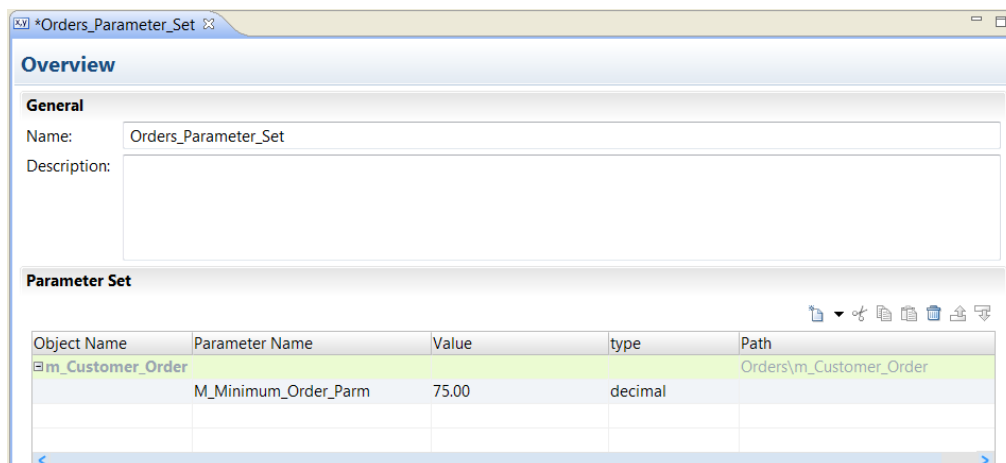


5. **パラメータの追加**ダイアログボックスで、[参照] をクリックし、セットに含める必要があるパラメータを含むマッピングまたはワークフローを見つけます。  
マッピングおよびワークフローのリストが表示されます。

6. マッピングまたはワークフローを選択し、[OK] をクリックします。  
マッピングまたはワークフローのパラメータのリストが表示されます。



7. パラメータセットに含めるパラメータを選択し、[OK] をクリックします。  
パラメータセットにマッピングまたはワークフロー名とパスが表示されます。選択した各パラメータがオブジェクトの下に表示されます。



8. ワークフローまたはマッピングにないパラメータを追加するには、マッピングまたはオブジェクト名を右クリックして [パラメータ] 挿入を選択します。

Developer tool によりワークフローまたはマッピングの下にパラメータが作成されます。パラメータ名、値、およびタイプを変更します。

**注:** パラメータセットを使用する前にマッピングまたはワークフローにパラメータを追加する必要があります。

## パラメータセットを使用したワークフローの実行

パラメータセットを指定してワークフローを実行するには、startWorkflow コマンドを使用します。-ps 引数でパラメータセット名を指定します。

パラメータセットを指定してワークフローをデプロイする場合、コマンドにパラメータセット引数を含めると実行時に異なるパラメータセットを使用できます。

例えば、次のコマンドでは、ワークフロー「myWorkflow」がパラメータセット「MyParameterSet」を指定して実行されます。

```
infacmd wfs startWorkflow -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -wf MyWorkflow -ps MyParameterSet
```

## パラメータファイル

パラメータファイルは、ユーザー定義パラメータおよびその割り当て値を列挙した XML ファイルです。パラメータファイルを使用すると、ワークフローを実行するたびにパラメータ値を変更することができます。

パラメータ値を使用して、ワークフローのプロパティまたはワークフローを実行するマッピングタスクに含まれるマッピングのプロパティを定義します。データ統合サービスでこれらの値を適用するには、コマンドラインからワークフローを実行するときにパラメータファイルを指定します。

パラメータファイルでマッピングパラメータおよびワークフローパラメータを定義できます。再利用可能オブジェクトパラメータを指定する場合は、再利用可能オブジェクトパラメータをマッピングパラメータとして公開します。パラメータファイルで、マッピングパラメータ値を指定します。

パラメータファイルでシステムパラメータの値を定義することはできません。

1 つのパラメータファイルで複数のワークフローのパラメータを定義できます。複数のパラメータファイルを作成し、ワークフローを実行するたびに異なるファイルを使用することもできます。データ統合サービスは、ワークフロー実行の開始時にパラメータファイルを読み込み、パラメータを解決します。

Developer tool からパラメータファイルをエクスポートできます。マッピングまたはワークフローの【**パラメータ**】タブからファイルをエクスポートします。Developer tool によって、マッピングまたはワークフローのパラメータとデフォルトのパラメータ値を含むパラメータファイルが生成されます。パラメータファイル名を指定し、ファイルの保存場所を選択できます。

**注:** パラメータファイルの構造は、マッピングでもワークフローでも同じです。1 つのパラメータファイルでデプロイ済みのマッピングとデプロイ済みのワークフローのパラメータを定義できます。

コマンドラインからワークフローで使用されるパラメータとデフォルト値を一覧表示することもできます。コマンドライン出力は、パラメータファイルのテンプレートとして使用できます。

コマンドラインからワークフローを実行し、パラメータファイルを適用します。

## ワークフローのパラメータファイルの例

ListWorkflowParams infacmd を実行してワークフローのパラメータファイルを作成できます。ワークフローのパラメータファイルにはマッピングパラメータは含まれません。マッピングパラメータを手動でファイルに追加できます。

```
<?xml version="1.0" encoding="UTF-16LE"?>
<root xmlns="http://www.informatica.com/Parameterization/1.0"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema" ><!--Specify deployed application specific parameters here.-->
  <!--^M
    <application name="a2">
      <workflow name="w2"/>
    </application>
  </-->
```



```

</application>--><project name="Orders">
  <folder name="integer_op">
    <folder name="Workflows">
      <workflow name="wf_Orders">
        <parameter name="wfStringParam">verboseData</parameter>
        <parameter name="wfConnectionParam">OracleDB</parameter>
      </workflow>
    </folder>
  </folder>
</project>
</root>

```

## パラメータファイルの例

次の例では、マッピングで作成可能なパラメータファイルの例を示します。

```

<?xml version="1.0" encoding="UTF-8">
<root xmlns="http://www.informatica.com/Parameterization/1.0"
xmlns:xsl="http://www.w3.org/2001/XMLSchema">

  <project name="sample_project">

    <mapping name="m_myMapping">

      <parameter name="srcConnect">RelationalConnect</parameter>

      <parameter name="srcDirectory">C:\Srcfiles</parameter>

      <parameter name="wrkDirectory">C:\TempFiles</parameter>

    </mapping>

  </project>

</root>

```

## パラメータファイルに関するルールとガイドライン

パラメータファイルを作成するときは、特定のルールとガイドラインが適用されます。

パラメータファイルを作成する際のルールは次のとおりです。

- パラメータファイルでマッピングおよびワークフローパラメータを参照できます。再利用可能なトランスフォーマー、マッピング、またはデータオブジェクトパラメータは参照できません。再利用可能オブジェクトパラメータを参照するには、再利用可能オブジェクトパラメータをマッピングパラメータとして公開します。パラメータファイルで、マッピングパラメータ値を指定します。
- アプリケーション要素には、アプリケーション実行時に指定したアプリケーションのみに適用されるマッピングまたはワークフローパラメータが含まれます。Developer tool からパラメータファイルを使用してマッピングを実行する場合、アプリケーションとしてマッピングをデプロイしないでください。また、アプリケーション要素を指定しないでください。プロジェクト要素内のマッピングを指定します。
- パラメータ値は空にできません。例えば、パラメータファイルに以下のエントリがある場合、データ統合サービスはワークフローしません。  

```
<parameter name="Param1"> </parameter>
```
- 要素内のアーティファクト名は大文字と小文字が区別されません。したがって、データ統合サービスは  

```
<parameter name="SrcDir">
```

と  

```
<parameter name="Srcdir">
```

を同じアプリケーションとして解釈します。
- 参照テーブルを特定するパラメータは、リポジトリフォルダパスでフォルダ名を区切るのにフォワードスラッシュ (/) を使用する必要があります。

## パラメータファイルのエクスポート

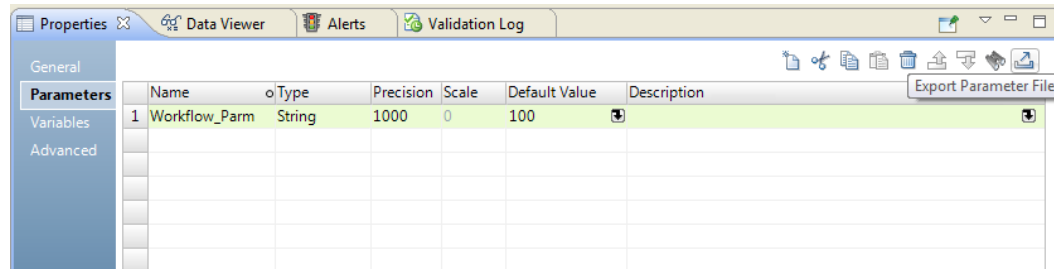
Developer tool から、マッピングパラメータファイルまたはワークフローパラメータファイルのエクスポートできます。Developer tool でパラメータを定義してから、そのパラメータをファイルにエクスポートします。Developer tool では、.XML 形式でパラメータファイルを作成します。

マッピングパラメータまたはワークフローパラメータを含むパラメータファイルのエクスポートできます。マッピングの **【パラメータ】** タブまたはワークフローの **【パラメータ】** タブから、パラメータをエクスポートできます。Developer tool によって、**【パラメータ】** タブからすべてのパラメータがエクスポートされます。

パラメータファイルのエクスポートするには、次の手順を実行します。

1. マッピングまたはワークフローのパラメータおよびパラメータのデフォルトを定義します。
2. マッピングまたはワークフローの **【プロパティ】** の **【パラメータ】** タブで、**【パラメータファイルのエクスポート】** オプションをクリックします。
3. パラメータファイルの名前を入力して、ファイルを配置する場所を参照します。
4. **【保存】** をクリックします。

次の図は、ワークフローの **【パラメータ】** タブの **【パラメータファイルのエクスポート】** オプションを示しています。



パラメータファイルのエクスポートすると、Developer tool によって、マッピングパラメータまたはワークフローパラメータのいずれかを含むパラメータファイルが作成されます。Developer tool では、マッピングパラメータとワークフローパラメータを同じファイルにエクスポートしません。

例えば、ワークフローパラメータ Workflow\_Parm をエクスポートすると、Developer tool では次のパラメータファイルを作成します。

```
<?xml version="1.0" encoding="UTF-16LE"?>
<root version="2.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema" xmlns="http://www.informatica.com/Parameterization/1.0">
  <project name="Orders">
    <workflow name="Customer_Workflow">
      <parameter name="Workflow_Parm">100</parameter>
    </workflow>
  </project>
</root>
```

## infacmd ms ListMappingParams からのパラメータファイルの作成

infacmd wfs ListWorkflowParams コマンドを実行して、デプロイ済みアプリケーションのワークフローで使用するパラメータと各パラメータのデフォルト値をリスト表示します。このコマンドの出力を使用して、パラメータファイルを作成します。

1. infacmd wfs ListWorkflowParams コマンドを実行して、ワークフローで使用するパラメータと各パラメータのデフォルト値をリスト表示します。  
-o 引数は、コマンド出力を XML ファイルに出力します。

例えば、次のコマンドでは、ワークフロー「MyWorkflow」のパラメータがファイル「MyOutputFile.xml」にリストされます。

```
infacmd wfs ListWorkflowParams -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -wf MyWorkflow -o MyOutputFile.xml
```

データ統合サービスは、すべてのマッピングパラメータをデフォルト値とともにリスト表示します。

2. -o 引数を指定しない場合、コマンド出力を XML ファイルにコピーしてファイルを保存できます。
3. XML ファイルを編集し、パラメータのデフォルト値を、ワークフローの実行時に使用する値に置き換えます。
4. XML ファイルを保存します。

## パラメータファイルを使用したワークフローの実行

コマンドラインからパラメータファイルを使用してマッピングを実行するには、マッピングをアプリケーションとしてデプロイする必要があります。マッピングを実行してパラメータファイルを指定します。infacmd wfs StartWorkflow コマンドを使用します。-pf 引数でパラメータファイル名を指定します。

例えば、次のコマンドでは、ワークフロー「MyWorkflow」がパラメータファイル「MyParamFile.xml」を使用して実行されます。

```
infacmd wfs StartWorkflow -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -a MyApplication -wf MyWorkflow -pf MyParamFile.xml
```

パラメータファイルを使用してマッピングを実行し、そのファイルが無効であった場合、データ統合サービスは、ワークフローを失敗にします。パラメータファイルが見つからないまたはアクセスできない場合、データ統合サービスは、ワークフローを失敗にします。

infacmd を使用してパラメータセットを使用する方法の詳細については、『*Informatica コマンドリファレンス*』を参照してください。

## 第 4 章

# クラスタタスク

## クラスタタスクの概要

クラスタタスクは、クラウドプラットフォーム上で計算クラスタを作成および削除できるワークフロータスクです。

クラスタの作成タスクを使用して、計算クラスタを作成、構成、起動します。クラスタの削除タスクを使用して、ワークフロー内のマッピングタスクとその他のタスクが完了した後に、同じクラスタを削除します。

クラスタワークフローは、クラスタ作成タスクと 1 つ以上のマッピングタスクが含まれるワークフローです。クラスタワークフローを実行すると、クラスタの作成タスクはコマンドを発行し、指定したサイズおよびタイプのクラスタを作成します。ワークフロー内のマッピングタスクとその他のタスクは、同じクラスタまたは指定した別のクラスタ上で実行できます。すべてのタスクが完了したらクラスタを終了する、クラスタの削除タスクを含めることができます。クラスタの削除タスクを含めない場合、クラスタは実行されたままになります。

クラスタワークフローの詳細については、*Data Engineering Integration* ユーザーガイドを参照してください。

## クラスタの作成タスク

クラスタの作成タスクには、ディストリビューションがマスタノードおよび作業ノードを持つクラスタを作成するために必要なすべての設定が含まれます。また、Hadoop 接続または Databricks 接続とクラウドプロビジョニング設定への参照も含まれます。

クラスタワークフローを作成するときは、クラスタの作成タスクをワークフローエディタにドラッグし、タスクのプロパティを設定します。

クラスタワークフローには、クラスタの作成タスクが 1 つのみ存在します。

クラウドのプラットフォームタイプに基づいて、詳細プロパティを設定します。

## クラスタの作成タスクの全般プロパティ

以下の表に、クラスタの作成タスクについて設定する全般プロパティを示します。

プロパティ	説明
名前	タスクの名前。
説明	オプション。タスクの説明。
接続名	ワークフローと一緒に使用するクラウドプロビジョニング設定の名前。
接続タイプ	以下のいずれかの Hadoop ディストリビューションを選択します。 <ul style="list-style-type: none"><li>- Amazon EMR</li><li>- HDInsight</li><li>- Azure Databricks</li><li>- AWS Databricks</li></ul> デフォルトは Amazon EMR です

## クラスタの作成タスクの出力

クラスタの作成タスクの出力プロパティを入力します。

クラスタ識別子プロパティがデフォルト値の AutoDeployCluster に設定されていることを確認してください。

## Amazon EMR の詳細プロパティ

Amazon EMR クラスタの詳細プロパティを設定します。

### 全般オプション

次の表に、EMR クラスタに対して設定できる全般オプションを示します。

プロパティ	説明
クラスタ名	作成するクラスタの名前。
リリースバージョン	クラスタ上で実行する EMR バージョン。 バージョンを指定する AWS バージョンタグ文字列を入力します。例: emr-5.8.0 デフォルトはサポートされている最新バージョンです。
接続名	クラスタワークフローと一緒に使用するために設定した Hadoop 接続の名前。
S3 ログ URI	オプション。クラスタ作成のためのログの S3 の場所。形式: s3://<bucket name>/<folder name> 場所を指定しない場合、クラスタログは保存されません。

## マスタインスタンスグループのオプション

次の表に、EMR クラスタに対して設定できるマスタインスタンスグループのオプションを示します。

プロパティ	説明
マスタインスタンス タイプ	マスタノード EC2 インスタンスタイプ。 任意の使用できる EC2 インスタンスタイプを指定できます。 デフォルトは m4.4xlarge です。
マスタインスタンスの 最大スポット価格	マスタノードの最大スポット価格。このプロパティを設定すると、マスタインスタンスグループの購入オプションがオンデマンドではなくスポットに変更されます。

## コアインスタンスグループのオプション

次の表に、EMR クラスタに対して設定できるコアインスタンスグループのオプションを示します。

プロパティ	説明
コアインスタンス タイプ	コアノード EC2 インスタンスタイプ。 任意の使用できる EC2 インスタンスタイプを指定できます。 デフォルトは m4.4xlarge です。
コアインスタンス 数	クラスタで作成するコア EC2 インスタンスの数。 デフォルトは 2 です。
コアインスタンス 最大スポット 価格	コアノードの最大スポット価格。このプロパティを設定すると、コアインスタンスグループの購入オプションがオンデマンドではなくスポットに変更されます。
コア自動スケー リングポリシー	オプション。コアインスタンスの自動スケーリングポリシー。ここにポリシー JSON 文を入力するか、JSON 文が格納されているファイルへのパスを入力します。 形式: file:\\<path_to_policy_config_file>

## タスクインスタンスグループのオプション

次の表に、EMR クラスタに対して設定できるタスクインスタンスグループのオプションを示します。

プロパティ	説明
タスクインスタンス タイプ	タスクノード EC2 インスタンスタイプ。 任意の使用できる EC2 インスタンスタイプを指定できます。 デフォルトは m4.4xlarge です。
タスクインスタンス 数	クラスタで作成するタスク EC2 インスタンスの数。 デフォルトは 2 です。

プロパティ	説明
タスクインスタンスの最大スポット価格	タスクノードの最大スポット価格。このプロパティを設定すると、タスクインスタンスグループの購入オプションは、オンデマンドではなくスポットに変更されます。
タスク自動スケーリングポリシー	オプション。タスクインスタンスの自動スケーリングポリシー。ここにポリシー JSON 文を入力するか、JSON 文が格納されているファイルへのパスを入力します。 形式: file: \\<path_to_policy_config_file>

## 追加オプション

次の表に、EMR クラスタに対して設定できる追加オプションを示します。

プロパティ	説明
ルートデバイス EBS ボリュームのサイズ (GB)	EBS ルートデバイスボリュームの GB 数。10～100 の範囲で値を入力してください。 デフォルトは 10 です。
タグ	オプション。クラスタ EC2 インスタンスにプロパゲートするタグ。 EC2 インスタンスの特定を補助するタグ。 形式: TagName1=TagValue1, TagName2=TagValue2
ブートストラップアクション	オプション。EC2 インスタンスの実行後、アプリケーションのインストール前に実行するアクション。 ここに JSON 文を入力するか、JSON 文が格納されているファイルへのパスを入力します。形式: file: \\<path_to_policy_config_file>
カスタム AMI ID	オプション。カスタム Amazon Linux Amazon Machine Image (AMI) の ID。AWS コンソールから値をコピーします。
セキュリティ設定	<p>オプション。クラスタ上での認証および暗号化のためのセキュリティ設定の名前。</p> <p>Amazon EMR は、サーバーサイドの暗号化 (SSE) とクライアントサイドの暗号化 (CSE) の設定をサポートします。</p> <p>以下の停止時のセキュリティ設定が使用可能です。</p> <ul style="list-style-type: none"> <li>- Amazon S3 で管理されたキーによる SSE (SSE-S3)</li> <li>- AWS KMS で管理されたキーによる SSE (SSE-KMS)</li> <li>- AWS KMS で管理されたキーによる CSE (CSE-KMS)</li> <li>- カスタムの CSE 設定*</li> </ul> <p>以下の転送時のセキュリティ設定が使用可能です。</p> <ul style="list-style-type: none"> <li>- PEM</li> <li>- カスタムの転送時の設定*</li> </ul> <p>ローカルディスクセキュリティにはカスタムの AMI も使用できます。</p> <p>* カスタムのセキュリティ設定を使用するには、<i>.jar</i> ファイルをデータ統合サービスマシンに手動でコピーします。</p>

プロパティ	説明
アプリケーション	<p>オプション。AWS によりインストールされるデフォルトアプリケーションに追加するアプリケーション。</p> <p>AWS によって、EMR クラスタの作成時に特定のアプリケーションがインストールされます。さらに、追加のアプリケーションを指定できます。追加のアプリケーションをドロップダウンリストから選択します。</p> <p>このフィールドは、AWS EMR クラスタ作成ウィザードのソフトウェア設定リストと同じです。</p>
ソフトウェア設定	<p>オプション。クラスタにインストールされているアプリケーションに適用するカスタム設定。このフィールドは、AWS クラスタ作成ウィザードのソフトウェア設定の編集フィールドと同じです。この方法を使用して、クラスタ上のソフトウェア設定を変更できます。</p> <p>ここに設定 JSON 文を入力するか、JSON 文が格納されているファイルへのパスを入力します。形式: file:\\&lt;path_to_custom_config_file&gt;</p>
手順	<p>オプション。クラスタ作成後に実行するコマンド。例えば、Linux コマンド、または HDFS コマンドや Hive Hadoop コマンドを実行するために使用できます。</p> <p>このフィールドは、AWS クラスタ作成ウィザードの手順の追加フィールドと同じです。</p> <p>ここにコマンド文を入力するか、JSON 文が格納されているファイルへのパスを入力します。形式: file:\\&lt;path_to_command_file&gt;</p>

## Azure Databricks の詳細プロパティ

Azure Databricks のクラスタの作成タスクの詳細プロパティの全般オプションと詳細オプションを設定します。

### 全般オプション

次の表に、Databricks クラスタに対して設定できる全般オプションを示します。

プロパティ	説明
クラスタ名	作成するクラスタの名前。
Databricks Runtime Version	<p>クラスタ上で実行する Databricks バージョン。</p> <p>デフォルトは、最新サポートバージョンです。</p> <p>バージョンを手動で入力するには、(タスク入力に割り当てる)を選択します。Developer tool が、<i>Databricks Runtime Version</i> というクラスタ作成タスク入力を作成します。この入力の値を、Databricks バージョンタグ文字列に設定します。</p> <p>例えば、Databricks バージョン 5.5 を使用するには、この値を 5.5.x-scala2.11 に設定します。</p>
Python バージョン	<p>Python トランスフォーマーメーションを含める場合に実行する Python のバージョン。</p> <p>デフォルトは、最新サポートバージョンです。</p>
ウォームプール	<p>ウォームプールを使用した Databricks クラスタの作成を有効にするには、このオプションをオンにします。</p> <p>デフォルトはオフ (false) です。</p>



プロパティ	説明
プール ID	ウォームプールリソースを識別する DatabricksInstancePoolId プロパティの値。 このプロパティの値を取得するには、Databricks ワークスペースのウォームプールに移動して、[設定] > [タグ] を参照します。 [ウォームプール] オプションがオンになっている場合は必須です。
ドライバの種類	ドライバノードで使用するノードのタイプ。 デフォルトは、ワーカーの種類 ID です。
ワーカーの種類	作業ノードで使用するノードのタイプ。
ワーカー数	クラスタで作成する作業ノードの数。クラスタの自動スケーリングを設定している場合は、このプロパティは無視されます。 デフォルトは 1 です。
自動スケーリング	負荷に基づいて、作業ノードの数を自動的にスケーリングします。
最小ワーカー数	クラスタの自動スケーリングを設定しているときに使用する、作業ノードの最小数。 デフォルトは 0 です。
最大ワーカー数	クラスタの自動スケーリングを設定しているときに使用する、作業ノードの最大数。 デフォルトは 1 です。

## 詳細オプション

環境変数や自動終了などの詳細オプションを設定します。

次の表に、Azure Databricks クラスタに対して設定できる詳細オプションを示します。

プロパティ	説明
自動終了	クラスタの自動終了を有効にします。
自動終了時間	指定された分数、非アクティブだった後にクラスタを終了します。10~10,000 の範囲で値を入力してください。これを設定しない場合、または 0 に設定した場合、クラスタは自動終了しません。
クラスタログ設定	長期ストレージ用のログを記録する場所。設定した場合、Databricks Spark エンジン は 5 分間隔でログを記録します。 DBFS へのパスを指定します。
初期化スクリプト	初期化スクリプトを保存する場所。複数の場所を入力できます。スクリプトは、設定した順序で順番に実行されます。追加の Python ライブラリをインストールする必要がある場合は、このプロパティで初期化スクリプトファイルの場所を指定します。 次の形式を使用します。 dbfs:/<path to init script>,dbfs:/<path to init script>

プロパティ	説明
クラスタタグ	<p>トレースの目的でリソースに割り当てることができるラベル。  &lt;key1&gt;=&lt;value1&gt;,&lt;key2&gt;=&lt;value2&gt;の形式でキーと値のペアを入力します。キーと値のペアを含むローカルファイルへのパスを指定することもできます。  次の形式を使用します。  file: \\&lt;file path&gt;</p>
Spark 設定	Databricks Spark エンジンのパフォーマンス設定。key1='value1' key2='value2'の形式でキーと値のペアを入力します。キーと値のペアを含むファイルへのパスを指定することもできます。
環境変数	Databricks Spark エンジンに対して設定できる環境変数。key1='value1' key2='value2'の形式でキーと値のペアを入力します。

## AWS Databricks の詳細プロパティ

AWS Databricks クラスタの詳細プロパティを設定します。

### 全般オプション

次の表に、Databricks クラスタに対して設定できる全般オプションを示します。

プロパティ	説明
クラスタ名	作成するクラスタの名前。
Databricks Runtime Version	<p>クラスタ上で実行する Databricks バージョン。  デフォルトは、最新サポートバージョンです。  バージョンを手動で入力するには、(タスク入力に割り当てる)を選択します。  Developer tool が、<i>Databricks Runtime Version</i> というクラスタ作成タスク入力を作成します。この入力の値を、Databricks バージョンタグ文字列に設定します。  例えば、Databricks バージョン 5.5 を使用するには、この値を 5.5.x-scala2.11 に設定します。</p>
Python バージョン	Python トランスフォーメーションを含める場合に実行する Python のバージョン。 デフォルトは、最新サポートバージョンです。
ウォームプール	<p>ウォームプールを使用した Databricks クラスタの作成を有効にするには、このオプションをオンにします。  デフォルトはオフ (false) です。</p>
プール ID	<p>ウォームプールリソースを識別する DatabricksInstancePoolId プロパティの値。  このプロパティの値を取得するには、Databricks ワークスペースのウォームプールに移動して、[設定] &gt; [タグ] を参照します。  [ウォームプール] オプションがオンになっている場合は必須です。</p>
ドライバの種類	<p>ドライバノードで使用するノードのタイプ。  デフォルトは、ワーカーの種類 ID です。</p>
ワーカーの種類	作業ノードで使用するノードのタイプ。

プロパティ	説明
ワーカー数	クラスタで作成する作業ノードの数。クラスタの自動スケーリングを設定している場合は、このプロパティは無視されます。 デフォルトは 1 です。
自動スケーリング	負荷に基づいて、作業ノードの数を自動的にスケーリングします。
最小ワーカー数	クラスタの自動スケーリングを設定しているときに使用する、作業ノードの最小数。 デフォルトは 0 です。
最大ワーカー数	クラスタの自動スケーリングを設定しているときに使用する、作業ノードの最大数。 デフォルトは 1 です。

## 詳細オプション

自動終了やオンデマンドインスタンスなどの詳細オプションを設定します。

次の表に、AWS Databricks クラスタに対して設定できる詳細オプションを示します。

プロパティ	説明
ローカルストレージの自動スケーリングの有効化	Databricks で、ワーカーノード上の使用可能なディスク容量を監視し、追加の EBS ボリュームを自動的に追加できるようにします。
EBS ボリュームタイプ	Databricks がクラスタノードに追加できるボリュームのタイプ。 このプロパティは、自動スケーリングを有効にするときに設定します。
ボリューム数	各インスタンスに対してプロビジョニングするボリューム数。0～10 の範囲で値を入力してください。 このプロパティは、自動スケーリングを有効にし、EBS ボリュームタイプを設定するときに設定します。
GB 単位のサイズ	各 EBS ボリュームのサイズ（ギガバイト単位）。 このプロパティは、自動スケーリングを有効にし、EBS ボリュームタイプを設定するときに設定します。
自動終了	クラスタの自動終了を有効にします。
自動終了時間	指定された分数、非アクティブだった後にクラスタを終了します。10～10,000 の範囲で値を入力してください。これを設定しない場合、または 0 に設定した場合、クラスタは自動終了しません。
オンデマンド/スポット構成	オンデマンドノードの数。0～[全般オプション] で設定したワーカーノード数の範囲で値を入力します。残りのワーカーノードはすべてスポットインスタンスになります。 オンデマンドノードは常に使用可能です。スポットインスタンスは、使用不可になると、実行中のジョブを停止する場合があります。ドライバノードは常にオンデマンドノードです。 このプロパティは、[オンデマンドにフォールバックするスポット] を有効にするときに設定します。 デフォルトは 1 です。

プロパティ	説明
オンデマンドにフォールバックするスポット	オンデマンドインスタンスをフォールバックとして使用できるようにします。 スポットインスタンスを使用しているときに、スポットインスタンスの市場価格がスポットの入札価格を上回った場合、AWS はそのスポットインスタンスを終了します。このプロパティを有効にすると、スポットインスタンスが終了したときにオンデマンドインスタンスが代わりに使用されます。
可用性ゾーン	AWS クラスタの可用性ゾーン。 デフォルトは us-east-1e です。
スポットの入札価格	スポットインスタンスに入札したオンデマンドインスタンス価格の最大割合 (%)。 スポットインスタンスは、オンデマンド価格の割合 (%) として価格付けされ、常に使用可能なわけではありません。 スポットインスタンスの市場価格がここで設定した入札価格を上回ったときに、 <b>【オンデマンドにフォールバックするスポット】</b> を有効にしていない場合、AWS はスポットインスタンスを終了します。 デフォルトは 100% です。
IAM ロール ARN	AWS IAM (Identity and Access Management) ロールに対応するインスタンスプロファイルの ARN (Amazon Resource Name)。AWS コンソールから以下の形式で値をコピーします。 <code>arn:aws:iam::&lt;account-id&gt;:instance-profile/&lt;role-name&gt;</code> IAM ロールがあると、Databricks クラスタからデータにアクセスできます。Administrator ツールで新しい IAM ロールを追加します。
Spark 設定	Databricks Spark エンジンのパフォーマンス設定。key1='value1' key2='value2' の形式でキーと値のペアを入力します。キーと値のペアを含むファイルへのパスを指定することもできます。
環境変数	Databricks Spark エンジンに対して設定できる環境変数。key1='value1' key2='value2' の形式でキーと値のペアを入力します。
クラスタタグ	トレースの目的でリソースに割り当てることができるラベル。 <key1>=<value1>,<key2>=<value2> の形式でキーと値のペアを入力します。キーと値のペアを含むローカルファイルへのパスを指定することもできます。 次の形式を使用します。 <code>file:\&lt;&gt;</code>
SSH パブリックキー	SSH を有効にしている場合にドライバおよびワーカーインスタンスにログインするための SSH パブリックキー。Databricks コンソールから値をコピーします。
クラスタログ設定	長期ストレージ用のログを記録する場所。設定した場合、Databricks Spark エンジン は 5 分間隔でログを記録します。 DBFS へのパスを指定します。
初期化スクリプト	初期化スクリプトを保存する場所。複数の場所を入力できます。スクリプトは、設定した順序で順番に実行されます。追加の Python ライブラリをインストールする必要がある場合は、このプロパティで初期化スクリプトファイルの場所を指定します。 次の形式を使用します。 <code>dbfs:/&lt;path to init script&gt;,dbfs:/&lt;path to init script&gt;</code>

# クラスタの削除タスク

クラスタの削除タスクは、クラウドプラットフォーム上で Hadoop クラスタを終了して削除します。

クラスタワークフローにクラスタの削除タスクを追加しない場合、クラスタはワークフローの終了時に実行されたままになります。

作成され、ジョブが完了したときに削除されるクラスタは、一時クラスタと呼ばれます。

クラスタの削除タスクに次のプロパティを設定します。

- 全般プロパティで、必要に応じてクラスタの削除タスクの名前を変更します。
- 詳細プロパティで、タスクのリカバリ戦略プロパティを [タスクの再開] に設定します。

## 第 5 章

# コマンドタスク

この章では、以下の項目について説明します。

- [コマンドタスクの概要, 62 ページ](#)
- [コマンドの構文, 62 ページ](#)
- [コマンドタスクの入力, 63 ページ](#)
- [コマンドタスクの出力, 64 ページ](#)
- [コマンドタスクの詳細プロパティ, 65 ページ](#)
- [コマンドタスクのトラブルシューティング, 65 ページ](#)

## コマンドタスクの概要

コマンドタスクは、ワークフローの実行中に 1 つのシェルコマンドを実行するか外部の実行可能プログラムを起動するタスクです。

シェルコマンドを指定して、拒否ファイルの削除、ファイルのコピー、ターゲットファイルのアーカイブなどを行うことができます。コマンドではワークフローのパラメータや変数を使用することができます。

ワークフローを実行すると、コマンドタスクのパラメータおよび変数に入力データが渡され、その入力データを使用してコマンドが実行されます。コマンドタスクの実行が完了すると、出力データがワークフローの変数に渡されます。

コマンドタスクを設定するときは、実行するコマンド、タスクに必要な入力データ、およびタスクからワークフローに渡す出力データを指定します。さらに、コマンドの実行時にタスクで使用する詳細プロパティを設定します。

## コマンドの構文

コマンドの構文は、Data Integration Service を UNIX で実行する場合と Windows で実行する場合で異なります。

UNIX で Data Integration Service を実行する場合は、すべての有効な UNIX コマンドまたはシェルスクリプトを使用できます。ワークフローの実行中に次のコマンドが実行されます。<command>はコマンドタスクで入力したコマンドです。

```
/bin/sh -c "<command>"
```

Windows で Data Integration Service を実行する場合は、すべての有効な DOS またはバッチファイルを使用できます。ワークフローの実行中に次のコマンドが実行されます。<command>はコマンドタスクで入力したコマンドです。

```
cmd.exe /c "<command>"
```

例えば、シェルコマンドを使用してファイルを別のディレクトリにコピーすることができます。SALES\_ADJ ファイルをソースディレクトリ L からターゲットディレクトリ H にコピーする場合、Windows では次のシェルコマンドを入力します。

```
copy L:\sales\sales_adj H:\marketing\
```

UNIX で同じ操作を行う場合は、次のコマンドを入力します。

```
cp sales/sales_adj marketing/
```

コマンドを入力する際のルールとガイドラインは次のとおりです。

- コマンドに改行文字や復帰文字を含めることはできません。
- コマンドで外部の実行可能プログラムを実行するには、プログラムの完全修飾パスを入力します。例えば、myCustomApp.exe という名前のカスタムアプリケーションを実行するには、次のコマンドを使用します。

```
c:\myCustomApp.exe
```

コマンドタスクで外部プログラムを実行する場合、タスクはプログラムが終了するまで実行状態のままになります。

- 各コマンドタスクは Data Integration Service と同じ環境で実行されます。環境の設定を変更するには、設定を変更するスクリプトまたはバッチファイルを作成し、そのファイルに実行するコマンドを含め、次のコマンドを使用してファイルを実行します。

```
c:\mybatfile.bat
```

## コマンドのパラメータおよび変数

コマンドにワークフローのパラメータや変数を含めることができます。

ワークフローのパラメータや変数は、**【コマンド】** タブの **【入力】** タブで選択できます。また、所定の構文に従って、コマンド内にパラメータ名または変数名を入力することもできます。

例えば、次のコマンドでは、SourceDirectory という名前のワークフローパラメータを使用して、コピーするファイルのソースディレクトリを定義しています。

```
copy ${par:SourceDirectory} H:\marketing\
```

関連項目：

- [「式および文字列のパラメータ名」 \(ページ 41\)](#)
- [「式および文字列の変数名」 \(ページ 34\)](#)

## コマンドタスクの入力

コマンドタスクの入力は、ワークフローのパラメータや変数からコマンドタスクに渡されるデータです。

ワークフローのパラメータや変数のプロパティの値を定義するには、コマンドタスクの設定プロパティをタスク入力に割り当てます。コマンドタスクの設定プロパティは、**【詳細】** タブに表示されます。

例えば、ワークフローのすべてのコマンドタスクについて、作業ディレクトリを同じワークフローパラメータに割り当てるとします。各コマンドタスクで同じコマンド実行ファイルを実行する場合は、パラメータファイルで、開発環境の実行ファイルがあるディレクトリにワークフローパラメータの値を設定します。その後、ワ

ワークフローをプロダクション環境にデプロイするときに、各コマンドタスクでプロダクション環境にあるコマンド実行ファイルを使用する必要がある場合は、Developer ツールで各コマンドタスクを編集する代わりに、パラメータファイルでワークフローパラメータの値を変更します。

**注:** このコマンドタスクは作業ディレクトリパスを検証しません。ワークフローのコンテキスト内では、作業ディレクトリパスが無効であっても、コマンドタスクは正常に実行できます。

#### 関連項目：

- [「タスク入力へのワークフローパラメータの割り当て」](#) (ページ 40)
- [「タスク入力への変数の割り当て」](#) (ページ 33)

## コマンドタスクの出力

コマンドタスクの出力は、コマンドタスクからワークフロー変数に渡されるデータです。コマンドタスクの出力には、全般出力とタスク固有の出力が含まれます。

コマンドタスクを設定するときに、**[出力]** タブで、ワークフロー変数に割り当てるタスクの出力値を指定します。これにより、コマンドタスクの完了時に、コマンドタスクの出力値がワークフロー変数にコピーされます。

タスクが失敗した場合は、タスクの全般出力の値がワークフロー変数にコピーされ、タスク固有の出力値はコピーされません。タスクが強制終了された場合には、タスクの出力値はワークフロー変数にコピーされません。

例えば、コマンドタスクでは、コマンドが正常に実行されたかどうかを示す終了コードの出力値が生成されますが、この出力データにワークフローから直接アクセスすることはできません。このデータを以降のワークフローで使用するには、CommandExitCode という名前のワークフロー変数に終了コードの出力を割り当て、その CommandExitCode ワークフロー変数を条件付きシーケンスフローの式で使用します。これにより、コマンドが正常に実行された場合にワークフローの次のオブジェクトが実行されるようになります。

タスクの開始時刻、終了時刻、およびタスクの実行が成功したかどうかなど、全般出力にはすべてのタスクで生成される出力データが含まれます。

コマンドタスクの出力には、コマンドの実行時にコマンドタスクで生成されたデータが含まれます。

次の表に、コマンドタスクで生成される出力データを示します。

出力データ	データ型	説明
終了コード	Integer	コマンドから返される終了コード。コマンドが成功した場合は 0 が返されます。コマンドが失敗した場合は 0 以外の値が返されます。
標準エラー	String	コマンドから返される標準エラーメッセージ。デフォルトでは、最初の 1,024 文字が返されます。標準エラーの長さは、コマンドタスクの詳細設定プロパティで変更することができます。
標準出力	String	コマンドから返される標準出力。デフォルトでは、最初の 1,024 文字が返されます。標準出力の長さは、コマンドタスクの詳細設定プロパティで変更することができます。

#### 関連項目：

- [「タスク出力」](#) (ページ 28)
- [「タスク出力を使用した値の割り当て」](#) (ページ 31)



# コマンドタスクの詳細プロパティ

コマンドタスクの【詳細】タブには、タスクでのコマンドの実行に使用されるプロパティが表示されます。

以下のコマンドタスク詳細プロパティを設定します。

## タスクリカバリストラテジ

中断された、またはエラーが発生したタスクを、データ統合サービスが再実行するかスキップするかを決定します。ワークフローでリカバリが有効になっていない場合は、データ統合サービスはタスクリカバリストラテジを無視します。

デフォルトでは、タスクは再開リカバリストラテジを使用します。

## 作業ディレクトリ

コマンドを実行するディレクトリ。作業ディレクトリは、タスクに入力するコマンドで指定することもできます。両方でディレクトリを指定した場合は、このプロパティで指定したディレクトリではなく、コマンドで指定したディレクトリが使用されます。

作業ディレクトリの値を入力します。または、[作業ディレクトリ] プロパティをタスク入力に割り当てて、コマンドタスクの【入力】タブでワークフローのパラメータや変数に割り当てることができます。プロパティは string データ型です。

デフォルトはデータ統合サービスのホームディレクトリです。このコマンドタスクは作業ディレクトリパスを検証しません。

## 標準出力長

コマンドの標準出力で返される文字数。最大 32,768 文字です。

デフォルトは 1,024 文字です。

## 標準エラー長

コマンドの標準エラーで返される文字数。最大 4,096 文字です。

デフォルトは 1,024 文字です。

# コマンドタスクのトラブルシューティング

次に示す状況に対する解決方法は、コマンドタスクのトラブルシューティングで役に立つことがあります。

**コマンドタスクの実行したコマンドは成功しなかったものの、データ統合サービスが引き続きワークフロー内の次のオブジェクトを実行します。**

コマンドが失敗した場合や、無効な作業ディレクトリが指定された場合でも、コマンドタスクが正常に実行されることがあります。「成功」全般出力は、コマンドタスクが正常に実行されたかどうかを示します。終了コードのタスク出力は、コマンドが正常に実行されたかどうかを示します。コマンドタスクが成功したかどうかと、コマンドが終了コード 0 を返したかどうかは、どちらも条件付きシーケンスフローで確認することができます。

例えば、コマンドタスクが返す「成功」出力を取得するためのワークフローブール変数を作成します。そのコマンドが返す終了コードを取得する整数ワークフロー変数を作成します。コマンドタスクから出力側の条件付き出力シーケンスフローで、次の式を作成します。

```
$var:CommandTaskSuccessful = true AND $var:CommandExitCode = 0
```

これにより、前のコマンドタスクが正常に実行され、コマンドから終了コード 0 が返された場合に、次のタスクが実行されるようになります。

## 第 6 章

# ヒューマンタスク

この章では、以下の項目について説明します。

- [ヒューマンタスクの概要, 66 ページ](#)
- [ヒューマンタスクと例外データ管理, 67 ページ](#)
- [ヒューマンタスクのインスタンス, 68 ページ](#)
- [ヒューマンタスクのステップ, 68 ページ](#)
- [ヒューマンタスクのロール, 69 ページ](#)
- [ヒューマンタスクのプロパティ, 69 ページ](#)
- [ステップのプロパティ, 74 ページ](#)
- [ヒューマンタスクとワークフローの設定, 79 ページ](#)
- [ヒューマンタスクの設定, 79 ページ](#)
- [タスクインスタンスの作成, 80 ページ](#)

## ヒューマンタスクの概要

ヒューマンタスクは、ワークフローのデータに対して 1 人以上のユーザーが実行するアクションを定義します。Analyst ツールユーザーにマッピングタスクで実行するマッピングの出力の分析をさせる場合に作成します。

マッピングタスクでは、データ品質の問題が解決されていないデータセット内のレコードを識別できます。このレコードには、エラーや重複する情報が含まれている可能性があります。ヒューマンタスクは、Analyst ツールユーザーが作業可能な 1 つ以上のタスクインスタンスにレコードを再編成します。Analyst ツールユーザーはレコード内の問題を解決し、レコードのデータ品質ステータスを更新します。

ヒューマンタスクを設定する際に、1 つ以上のタスクステップを作成します。タスクステップは、タスクインスタンスのレコードに対してユーザーが実行する作業のタイプを表します。すべてのタスクデータを 1 人のユーザーに割り当てることも、複数のタスクインスタンスを作成して複数のユーザーが同時に別々のレコードを操作できるようにすることもできます。

ヒューマンタスクは、マッピングタスクが完了すると開始されます。マッピングタスクの *exceptionLoadCount* 出力値を使用して、ヒューマンタスクのデータが定義されます。すべてのタスクインスタンスが完了すると、ワークフローの次のステージにデータが渡されます。

# ヒューマンタスクと例外データ管理

ヒューマンタスクは、例外トランスフォーメーションを含むマッピングの出力を読み取ります。例外トランスフォーメーションは他のトランスフォーメーションの出力を分析し、データセットに含まれるレコードのデータ品質ステータスを検証します。マッピング作成者は、例外トランスフォーメーションを使用して、手動で処理する必要があるレコードを特定します。

例外トランスフォーメーションは、各レコードのデータ品質ステータスに基づいて、レコードを 1 つ以上のデータベーステーブルに書き込みます。データ品質ステータスを確認できないレコードに対して、ターゲットとしてテーブルが指定されます。ヒューマンタスクにおけるユーザーの役割は、そのテーブル内のレコードのデータ品質ステータスを確認することです。

マッピング作成者が作成した例外トランスフォーメーションを含むマッピングは、ワークフロー作成者によってワークフローのマッピングタスクに追加されます。ワークフローにヒューマンタスクを追加するときに、マッピングタスクの実行時に作成されるデータベーステーブルを読み取るように設定します。ヒューマンタスクを実行するユーザーは、それらのレコードを調べ、必要に応じて変更を行います。

次にユーザーは、次のいずれかの方法でレコードのステータスを更新します。

- レコードが有効な場合は、レコードのデータベースへの永続的な格納が確定されるよう、テーブルのメタデータを更新します。
- レコードが無効な場合は、ワークフローの以降のステージでレコードがデータベースから削除されるようにテーブルのメタデータを更新します。
- レコードのステータスを確定できない場合は、レコードをワークフローに戻してマッピングタスクで追加のさらに処理されるようにテーブルのメタデータを更新します。

## 例外データのタイプ

例外トランスフォーメーションは、データ品質ステータスを確認できないレコードを含むデータベーステーブルを生成します。ヒューマンタスクのユーザーは、各レコードを調べ、レコードに含まれる問題の解決を試みます。

レコードのデータ品質の問題としては、次のような問題が考えられます。

- レコードにエラーや空のセルが含まれている。ヒューマンタスクのユーザーは、レコードを調べ、正しく完全なデータでレコードを更新します。
- レコードが別のレコードと重複している。Analyst ツールでは、重複するレコードセットは**クラスタ**と呼ばれるグループで表示されます。ヒューマンタスクのユーザーは、クラスタを調べ、クラスタごとに優先するレコードのバージョンを 1 つ作成します。

ユーザーは、レコードまたはクラスタに次のステータスインジケータを適用できます。

- レコードまたはクラスタの問題が解決されており、レコードをデータベースに残すことができる。クラスタの場合は、優先レコードがテーブルに残され、不要な重複レコードは削除されます。
- レコードまたはクラスタの問題が解決されておらず、追加の処理が必要である。

レコードまたはクラスタは使用できないデータを含んでおり、テーブルから削除できる。

## Analyst ツール

Analyst ツールは Web ベースのアプリケーションで、ヒューマンタスクのレコードやクラスタを表示および更新できます。

Analyst ツールでは、受信トレイを使用して、割り当てられているヒューマンタスクのユーザーに通知します。ユーザーは Analyst ツールにログインして、**【マイタスク】** パネルからタスクを開きます。

Analyst ツールには、レコードやクラスタのデータを編集するオプションや、レコードやクラスタのステータスを更新するオプションが用意されています。タスクビューのメタデータカラムには、それぞれのレコードやクラスタのステータスインジケータが表示されます。

ユーザーが Analyst ツールでタスクを完了すると、タスクのレコードがヒューマンタスクの次のステップに渡されます。

## ヒューマンタスクのインスタンス

ワークフローにヒューマンタスクを追加するときは、タスクのインスタンスを 1 つ以上設定します。タスクインスタンスにユーザーを割り当てることで、データセットをユーザ間で分担して作業することができます。データ統合サービスはタスクインスタンスをワークフローの実行時に作成します。

ヒューマンタスクのインスタンスを複数作成する場合は、タスク分散オプションを使用します。タスク分散オプションにより、各タスクインスタンスのデータのサイズと構成が決まります。例えば、ヒューマンタスクの入力データセットに 1000 件のレコードが含まれている場合、タスクインスタンスを 5 つ作成して、各インスタンスに 200 件ずつレコードを含めることができます。

タスクインスタンスは、次の方法で作成できます。

- 作成するタスクインスタンスの数を指定する。データ統合サービスはタスクインスタンスに入力データセットを均等に配分します。
- タスクに含める行またはクラスタの数を指定する。データ統合サービスは指定した数の行またはクラスタを含むタスクインスタンスを作成します。タスクインスタンスの数は、入力データセットに含まれる行またはクラスタの数で決まります。
- 各ユーザーに割り当てるデータの行またはクラスタを指定する。入力データカラムを選択し、各タスクインスタンスを担当するユーザーを指定します。データ統合サービスは選択したカラムの値に基づいて、ユーザーに行またはクラスタを割り当てます。

1 つのヒューマンタスクに複数のステップを追加した場合、各タスクインスタンスに関連付けられたデータは、ヒューマンタスクの各ステップを実行していきます。ステップごとにタスクインスタンスを処理するユーザーを選択します。同じ組み合わせのユーザーをヒューマンタスクのすべてのステップに割り当てることも、それぞれのステップに異なるユーザーを割り当てることもできます。

## ヒューマンタスクのステップ

ヒューマンタスクには、ステップが 1 つ以上含まれます。ステップとは、Analyst ツールユーザーがヒューマンタスクを完了するために処理する必要のあるタスクインスタンスのブループリントです。ワークフローが実行されると、ステップは各タスクインスタンスに割り当てるデータを特定し、Analyst ツールユーザーがデータに実行する必要のある作業のタイプを定義します。

ヒューマンタスクに追加できるステップを次に示します。

- クラスタステップ。重複レコードクラスタを調べて重複レコードの値から優先レコードを作成する作業をユーザーに割り当てる場合に設定します。Analyst ツールは、重複を修正するタスクとしてクラスタステップで定義されたタスクを識別します。
- 例外ステップ。レコードのエラーを調べて修正する作業をユーザーに割り当てる場合に設定します。Analyst ツールは、例外を修正するタスクとして例外ステップで定義されたタスクを識別します。

- 確認ステップ。例外ステップまたはクラスタステップで実行した作業を確認する作業をユーザーに割り当てる場合に設定します。ステップの確認は強制ではありません。Analyst ツールは、重複を確認するタスクまたは例外を確認するタスクとして確認ステップで定義されたタスクを識別します。

例外ステップとクラスタステップはヒューマンタスクに一方のみを追加でき、両方を追加することはできません。クラスタステップと例外ステップのデータベーステーブルは構造が異なります。

任意の順番でステップを追加することができます。例えば確認ステップを、クラスタステップまたは例外ステップの後に追加しても、前に追加してもかまいません。

## ヒューマンタスクのロール

ヒューマンタスクを作成しステップとタスクインスタンスを設定するときに、タスクを処理することのできるユーザーまたはグループを指定することができます。ユーザーとグループに異なるロールを割り当てます。

以下のロールをユーザーおよびグループに割り当てることができます。

- タスクの実行者。Analyst ツールでタスクインスタンスを処理するように選択するユーザーまたはグループ。
- ビジネス管理者。ヒューマンタスクに（またはそのタスクのステップに）関連付けられたタスクインスタンスのステータスを管理するユーザーまたはグループタスクの実行者がスケジュールのタスクインスタンスを完了できない場合、ビジネス管理者がタスクを他のタスク実行者に割り当てなおします。

ビジネス管理者の指定は、ヒューマンタスクのレベルで行います。オプションとしてビジネス管理者をステップのレベルで指定することもできます。ヒューマンタスクで指定するビジネス管理者は、ヒューマンタスクの各ステップのビジネス管理者でもあります。ステップでビジネス管理者を定義する際、ビジネス管理者のステータスはそのステップ内のみに適用されます。

タスクの実行者の指定は、ヒューマンタスクのレベルとステップのレベルで行うことができます。タスクインスタンスに含まれるデータに基づいてユーザーまたはグループをタスクインスタンスに割り当てる必要がある場合は、ヒューマンタスクのレベルでタスク実行者を指定します。タスクの実行者をヒューマンタスクのレベルで指定する場合、タスクの実行者はヒューマンタスクの期間同じデータに対して処理を行います。ステップでタスクの実行者を定義する場合、タスクの実行者はそのステップ内のタスクに対してのみ処理を行います。

同じ名前のユーザーとグループをヒューマンタスクに割り当てることができます。例えば、グループドメイン/管理者とユーザードメイン/管理者を同じヒューマンタスクに割り当てることができます。

## ヒューマンタスクのプロパティ

ヒューマンタスクをワークフローに追加すると、ワークフローの【プロパティ】ビューにタスクのプロパティが表示されます。

以下のタブでオプションを設定します。

- [全般] タブ。ヒューマンタスクの名前と説明を設定します。
- [パーティシパント] タブ。タスクに参加できるビジネス管理者を指定します。
- [データソース] タブ。マッピングタスクの出力が格納されたデータベースのデータベース接続名とデータベースリソースを指定します。
- [タスク分散] タブ。タスクインスタンスのサイズと構成を設定します。

- [通知] タブ。ヒューマンタスクが完了したときに電子メールでの通知を受け取るユーザーを指定し、電子メールの文面を作成します。
- [入力] タブ。タスクの入力データを特定する変数を設定します。
- [出力] タブ。タスクの最終的なステータスをワークフローに渡す変数を設定します。
- [詳細] タブ。タスクのリカバリストラテジを設定します。

## [全般] タブ

[全般] タブでヒューマンタスクを指定します。ヒューマンタスクの名前と必要に応じて説明を入力します。ヒューマンタスクは [全般] タブから開くこともできます。

以下の表に、[全般] タブのオプションを示します。

プロパティ	説明:
名前	ヒューマンタスクの名前。
説明	オプション。ヒューマンタスクの説明。
タスク	ヒューマンタスクのリポジトリオブジェクトへのリンク [タスク] リンクをクリックして Developer ツールのヒューマンタスクを開きます。

## [パーティシパント] タブ

[パーティシパント] タブオプションを使用して、データ統合サービスでヒューマンタスク用に作成されたタスクインスタンスで、1名以上のビジネス管理者を指定します。ビジネス管理者は、Analyst ツールでタスクインスタンスを処理することもできます。

以下の表に、[パーティシパント] タブのオプションを示します。

プロパティ	説明
名前	ヒューマンタスクのビジネス管理者として指定するユーザーまたはグループの名前。 ビジネス管理者を選択するには、[選択] をクリックします。

ビジネス管理者はヒューマンタスクに割り当てることも、ヒューマンタスクの1つのステップに割り当てることもできます。ヒューマンタスクのビジネス管理者は、データ統合サービスによってヒューマンタスク用に作成されたタスクインスタンスで、タスク管理アクションを実行できます。ステップのビジネス管理者は、データ統合サービスによってステップ用に作成されたすべてのタスクインスタンスで、タスク管理アクションを実行できます。

ヒューマンタスクのビジネス管理者は、ヒューマンタスクに関連付けられたすべてのタスクインスタンスを1回の操作で完了することもできます。タスクを完了するには、ビジネス管理者はコマンドラインで `infacmd wfs completeTask` コマンドを実行します。

## [データソース] タブ

[データソース] タブでは、タスクデータが含まれるテーブルを特定するためのオプションを設定します。

以下の表に、[データソース] タブのオプションを示します。

プロパティ	説明
接続	マッピングタスクの出力が格納されたデータベースのデータベース接続名。
リソース	マッピングタスクの出力が格納されたデータベーステーブルの名前。タスクを実行するユーザーは、データベースに接続し、このテーブル内のデータに対して作業を行います。 テーブルが Oracle データベースまたは IBM DB2 データベース上にある場合、リソースとしてデータベーステーブルシノニムを選択することができます。

## [タスク分散] タブ

[タスク分散] タブでは、ヒューマンタスクのインスタンスを作成するためのオプションを設定します。タスク分散オプションは、タスクを複数のユーザーに割り当てる場合に使用します。同じサイズのタスクインスタンスを作成したり、カラムのデータに基づいてタスクインスタンスを作成したりできます。

以下の表に、[タスク分散] タブのオプションを示します。

プロパティ	説明
タスク分散の有効化	ヒューマンタスクのインスタンスを複数作成するためのオプションを有効にします。このオプションをオフにした場合は、ワークフローでのヒューマンタスクの処理で1つのインスタンスが処理されます。
項目数に基づいてタスクを分割する	指定した数のレコードまたはクラスタを含むタスクインスタンスを作成するか、指定した数のタスクインスタンスを作成します。
データ値に基づいてタスクを分割する	<p>選択したカラムの値に基づいてタスクインスタンスを作成します。</p> <p><b>[データソース]</b> タブで指定されたデータソースからカラムを選択します。ワークフローが実行されると、データ統合サービスでは、カラムの値が同じレコードのセットごとにタスクインスタンスを作成します。</p> <p>タスクインスタンスを処理できるユーザーまたはグループを特定し、各ユーザーまたはグループに関連付ける値を指定します。必要に応じて、各ユーザーまたはグループを数値または日付値の範囲に関連付けることができます。ワークフローでは、定義した関連付けに基づいて、タスクインスタンスがユーザーまたはグループに割り当てられます。</p> <p>次の方法で、タスクインスタンスを処理するユーザーまたはグループを指定します。</p> <ul style="list-style-type: none"><li>- グリッドにユーザーまたはグループを入力し、各ユーザーまたはグループに関連付けるカラム値を入力します。</li><li>- モデルリポジトリの参照テーブルから、ユーザー名またはグループ名とカラム値をインポートします。</li><li>- ローカルの区切りファイルから、ユーザー名またはグループ名とカラム値をインポートします。</li><li>- ユーザー名またはグループ名の一覧とデータ値の一覧を含む外部データベーステーブルに接続します。</li></ul> <p>外部データベースに接続すると、ワークフローの設定とは関係なく、ユーザー、グループ、および値の一覧を更新できます。データ統合サービスでは、ワークフローの実行時に、このテーブルを読み取ります。</p> <p>必要に応じて、タスク分散条件を満たさないレコードを含むタスクインスタンスを受信するユーザーまたはグループを指定できます。</p>

### 関連項目：

- [「タスクインスタンスの作成」 \(ページ 80\)](#)

## [通知] タブ

ヒューマンタスクが完了したときに、ユーザーまたはグループに通知するように [通知] タブのオプションを設定します。このワークフローは、指定したユーザーまたはグループに対し電子メールによる通知を送信します。

電子メールを送信するようにヒューマンタスクを設定する前に、管理者は Administrator ツールで電子メールサービスを有効にし、設定する必要があります。



以下の表に、[通知] タブのオプションを示します。

プロパティ	説明
タスクイベント	ユーザーまたはグループに通知することのできるステータスの変更の種類を一覧表示します。 ワークフローレベルでは完了オプションを選択することができます。
受信者	ヒューマンタスクが完了したときに通知する 1 つまたは複数のユーザーまたはグループを一覧表示します。
件名	電子メール通知の件名の内容を一覧表示します。件名は編集できます。
MIME タイプ	電子メールの本文の形式を指定します。次のいずれかの値を選択します。 - プレーン。メッセージをプレーンテキストで送信します。 - HTML。HTML 形式で本文を送信します。HTML の本文にはリストとハイパーリンクを含めることができます。 HTML ビューでは、デフォルトで HTML の基本構造を提供します。HTML を編集すると、デフォルトの構造を復元することはできません。
メッセージ	電子メールの本文を定義する HTML タグを表示します。メッセージの本文を<body>タグの間に入力します。
プレビュー	電子メールの本文を、受取人側で表示されるのと同じように表示します。

## [入力] タブ

ヒューマンタスクの入力は、ワークフロー変数からヒューマンタスクに渡されるデータです。

[入力] タブには、[処理された項目数] という 1 つのオプションが表示されます。このオプションは、*exceptionLoadCount* 変数を設定する場合に使用します。

*exceptionLoadCount* 変数には、[データソース] タブで指定したデータベーステーブルのレコードまたはクラスタの数が格納されます。この変数は、ヒューマンタスクで処理されるレコードまたはクラスタの数を示します。

**注:** ヒューマンタスクでは入力パラメータは設定しません。

### 関連項目：

- [「タスク入力への変数の割り当て」 \(ページ 33\)](#)

## [出力] タブ

ヒューマンタスクの出力は、ヒューマンタスクからワークフロー変数に渡されるデータです。ヒューマンタスクの出力には、全般出力が含まれます。タスクの開始時刻、終了時刻、およびタスクの実行が成功したかどうかなど、全般出力にはすべてのタスクで生成される出力データが含まれます。

ヒューマンタスクを設定するときに、ワークフロー変数に割り当てるタスクの出力値を [出力] タブで指定します。これにより、ヒューマンタスクの完了時に、ヒューマンタスクの出力値がワークフロー変数にコピーされます。

例えば、ヒューマンタスクでは、タスクの実行が開始された時刻を示す出力値が生成されますが、この出力データにワークフローから直接アクセスすることはできません。このデータを以降のワークフローで使用するには、HumanTaskStartTime という名前のワークフロー変数に開始時刻の出力を割り当て、その HumanTaskStartTime ワークフロー変数を条件付きシーケンスフローの式で使用します。これにより、ヒューマンタスクが指定の時刻よりも前に実行された場合にワークフローの次のオブジェクトが実行されるようになります。

関連項目：

- [「タスク出力」 \(ページ 28\)](#)
- [「タスク出力を使用した値の割り当て」 \(ページ 31\)](#)

## [詳細] タブ

ヒューマンタスクの [詳細] タブには、タスクのリカバリストラテジが含まれます。

ヒューマンタスクには再開リカバリストラテジが設定されています。再開オプションは読み取り専用です。データ統合サービスは、ワークフローのリカバリ時に常に中断されたヒューマンタスクを再開します。

ワークフローでリカバリが有効になっていない場合は、データ統合サービスはタスクリカバリストラテジを無視します。

## ステップのプロパティ

ヒューマンタスクにステップを追加するときは、ステップの作業を実行できるユーザーを指定し、ステップの継続時間を設定します。ステップの設定はプロパティビューで行います。

設定できるステップのオプションを次に示します。

- 全般。ステップの名前と説明を設定します。
- パーティシパント。データ統合サービスによってステップに作成されたタスクを処理するユーザーを指定します。
- 権限。タスクデータおよびメタデータの権限を指定します。権限は、ステップのタスクを処理するユーザーに適用されます。
- 設定。確認ステップのみ。ユーザーに確認される例外ステップまたはクラスタステップを指定します。
- タイムアウト。すべてのユーザーが現在のステップのタスクインスタンスを完了する必要がある期間を設定します。タイムアウトの値は、分数、時間数、および日数で設定します。タイムアウト期間は、ワークフローが実行された時点から計測されます。  
**注:** ヒューマンタスクでは、タイムアウト期間を使用して、ステップに関連付けられたすべてのタスクインスタンスの期限が計算されます。タスクインスタンスを実行するユーザーには、タイムアウト期間ではなく、期限の日時が表示されます。
- 通知。ステップに関連付けられたタスクインスタンスのステータスが変わるときに通知を受け取るユーザーを指定します。

## 全般オプション

全般オプションは、ヒューマンタスクのステップを指定するために設定します。ステップの名前を入力し、必要に応じて説明を入力します。

次の表に、ステップの全般オプションを示します。

プロパティ	説明:
名前	ヒューマンタスクのステップの名前。
説明	オプション。ステップの説明。

## 設定オプション

設定オプションを使用して、例外レコードを修正または確認するタスクの値をタスクの実行者が更新する方法を指定します。また、確認ステップで設定オプションを使用して、現在のステップに確認対象のタスクデータを渡すステップを指定します。

次の表に、ステップの設定オプションを示します。

プロパティ	説明
確認するステップ	現在のステップに確認対象のデータを渡すステップを指定します。 確認ステップのメニューからステップを選択します。
空のエラーセルを許可しない	タスクで例外として識別された null または空白のセルを、タスクの実行者が更新する必要があるかどうかを指定します。このオプションは、タスクを操作するすべてのユーザーに適用されます。  デフォルトでは、タスクの実行者はセルの値を空または null 値に変更し、null または空白のセルを例外として特定するタスクを完了します。タスクの実行者がすべての例外をデータ値で更新する必要があることを指定するには、このオプションを選択して、空のエラーセルを禁止します。  注: タスクの実行者に編集権限が与えられていないタスクでは、このオプションは適用されません。

## パーティシパントオプション

[パーティシパント] オプションを使用して、現在のステップで定義されているタスクインスタンスを処理できるユーザーまたはグループを指定します。タスクの実行者およびビジネス管理者の役割を果たすユーザーまたはグループを選択できます。

次の表に、ステップのパーティシパントオプションを示します。

プロパティ	説明
タスクの実行者	現在のステップでデータ統合サービスによって作成されたタスクインスタンスを処理できるユーザーまたはグループを指定します。 次の設定オプションを指定すると、データ統合サービスでステップのタスク実行者が無視されます。 <ul style="list-style-type: none"><li>- タスクインスタンスを定義して、例外または重複レコードを修正するステップを選択します。</li><li>- ヒューマンタスクでタスク分散プロパティを設定して、タスクをデータ値で分割します。 設定したタスク分散プロパティにより、ステップで定義されたタスクインスタンスを処理できるユーザーとグループが指定されます。</li></ul>
ビジネス管理者	現在のステップでデータ統合サービスによって作成されるタスクインスタンスについて、タスク管理操作を実行できるユーザーまたはグループを指定します。ビジネス管理者もタスクインスタンスを処理することができます。  ヒューマンタスクを設定するときに、ビジネス管理者を選択することもできます。ヒューマンタスク用に選択されたすべてのビジネス管理者は、ヒューマンタスクの各ステップで定義されたタスクインスタンスのビジネス管理者になります。

## 権限オプション

〔権限〕オプションを設定して、ユーザーが表示できるデータと、ユーザーが Analyst ツールで実行できるアクションのタイプを指定します。権限は、現在のステップが定義するタスクインスタンスを表示または編集できるすべてのユーザーに適用されます。

表示権限と編集権限を設定できます。表示権限によって、Analyst ツールに表示されるタスクインスタンスデータが定義されます。編集権限は、ユーザーがレコードデータまたはクラスタデータを更新するために実行できるアクションを定義します。権限を設定しても、ユーザーはレコード、クラスタ、またはタスクインスタンスのステータス情報を表示または更新できます。

例外ステップとクラスタステップではさまざまなタイプの権限がサポートされています。確認ステップで設定する権限は、前のステップが例外データステップかクラスタステップかによって異なります。

権限をステップに設定する場合は次のルールとガイドラインを検討します。

- 表示権限は編集権限よりも優先されます。ユーザーがタスクインスタンスのデータカラムを表示できない場合、そのユーザーはカラム内のデータを編集できません。
- ユーザーがタスクインスタンスのデータカラムを表示できない場合、そのユーザーは監査証跡内のカラムを表示できません。
- 設定するステップで、ヒューマンタスクの別のステップの出力を読み取る場合があります。Analyst ツールユーザーが前のステップで生成されたタスクインスタンスの適切なデータを表示および編集する権限を設定します。

### 表示権限

次の表に、各ステップで設定できる表示権限を示します。

権限値	説明	ステップのタイプ
すべてのデータを表示	Analyst ツールユーザーは、タスクインスタンスのデータをすべて表示できます。 デフォルトのオプションです。	すべてのステップのタイプ。
選択した列を表示	Analyst ツールユーザーは、タスクインスタンスで選択するカラムのデータを表示できます。	すべてのステップのタイプ。

### 権限の編集

次の表に、各ステップで設定できる編集権限を示します。

権限値	説明	ステップのタイプ
すべてのデータを編集	Analyst ツールユーザーは、タスクインスタンスの表示可能なデータをすべて編集できます。 例外データを読み取るステップのデフォルトオプションです。	例外ステップ 例外データの確認ステップ
選択した列を編集	Analyst ツールユーザーは、タスクインスタンスに 1 つ以上のカラムが表示される場合に選択するそれらのカラムのデータを編集できます。	すべてのステップのタイプ
例外データのみ編集	Analyst ツールユーザーは、ワークフローによってデータ品質の問題と見なされるタスクインスタンスの表示可能な値を編集できます。	例外ステップ 例外データの確認ステップ

権限値	説明	ステップのタイプ
すべてのアクションの実行	Analyst ツールユーザーは、タスクインスタンスの表示可能なデータのアクションをすべて実行できます。 クラスタデータを読み取るステップのデフォルトオプションです。	クラスタステップ クラスタデータの確認ステップ
クラスタアクションのみ実行	Analyst ツールユーザーは、タスクインスタンスの各クラスタの優先レコードを定義できます。Analyst ツールユーザーは、クラスタを作成してクラスタ間でレコードを移動できます。 Analyst ツールユーザーは、クラスタレコード内のデータ値を編集できません。	クラスタステップ クラスタデータの確認ステップ
確認アクションのみ実行	Analyst ツールユーザーは、タスクインスタンスのステータスとタスクインスタンスのレコードまたはクラスタを更新するアクションを実行できます。 Analyst ツールユーザーは、タスクインスタンスのデータ値や優先レコードを編集できません。	例外データの確認ステップ クラスタデータの確認ステップ

## 関連項目：

- [「ステップの権限オプションのルールおよびガイドライン」 \(ページ 83\)](#)

## タイムアウトオプション

[タイムアウト] オプションを使用して、ステップで定義されたタスクインスタンスをタスクの実行者が完了する必要がある期間を設定します。指定する期間内にタスクインスタンスが完了しなかった場合、Analyst ツールではそのようなタスクを期限切れタスクとして表示します。このワークフローでは期限切れのタスクを、ステップで指定したユーザーまたはグループに割り当てしなおすことができます。

次の表に、ステップのタイムアウトオプションを示します。

プロパティ	説明
継続時間	ステップに関連付けられたすべてのタスクインスタンスを完了しなければならない期間です。期間は、ヒューマンタスクがタスクインスタンスを作成したときを開始時とします。日数、時間、分で指定します。
名前	タスクの割り当てを受け取ることのできるユーザーまたはグループをリスト表示します。 複数のタスクが時間内に完了できないことが予測される場合、複数のユーザーかグループをリストに追加することを考慮してください。
タスクの再割り当て	再割り当てされたタスクをユーザーまたはグループが受け取るかどうか示します。 <b>【タスクの再割り当て】</b> オプションをクリアしたワークフローでは、期限切れになったタスクをそのユーザーまたはグループに再度割り当てることはありません。

## 通知オプション

ステップで定義されたタスクインスタンスのステータスが変更されたときに、ユーザーまたはグループに通知するように通知オプションを設定します。このヒューマンタスクは、指定したユーザーまたはグループに対し電子メールによる通知を送信します。

電子メールを送信するようにヒューマンタスクを設定する前に、管理者は Administrator ツールで電子メールサービスを有効にし、設定する必要があります。

次の表に、ステップの通知オプションを示します。

プロパティ	説明
タスクイベント	<p>ユーザーに通知することのできるステータス変化の種類をリスト表示します。以下の値のいずれかを選択できます。</p> <ul style="list-style-type: none"> <li>- 作成。ヒューマンタスクが、ステップに関連付けられたタスクインスタンスを作成する。</li> <li>- 完全。ユーザーがタスクインスタンスを完了します。</li> <li>- エスカレート。ユーザーまたはグループが、タスクインスタンスを時間どおりに完了しない。</li> <li>- 再割り当て。ビジネス管理者が、タスクインスタンスを他のユーザーに割り当てなおす。</li> </ul> <p><b>注:</b> 各タスクイベントに異なる通知を送信するように通知オプションを設定できます。例えば、ヒューマンタスクでタスクインスタンスが作成される場合に特定のユーザーに通知し、タスクインスタンスが完了した場合に別のユーザーに通知することができます。また、タスクイベントごとに異なる電子メールの件名とメッセージを入力することもできます。</p>
受信者	<p>選択したステータスの変更について通知する 1 人以上のユーザーを一覧表示します。</p> <p>タスクインスタンスのステータスが変更されたときに、選択した受信者に加えて、タスク所有者に通知するオプションを選択するか、オフにすることができます。このオプションは、1 人のユーザーがタスクインスタンスを所有している場合に適用されます。タスク所有者に通知するオプションを選択すると、必要に応じて「受信者」フィールドを空のままにできます。</p>
件名	電子メール通知の件名の内容を一覧表示します。件名は編集できます。
MIME タイプ	<p>電子メールの本文の形式を指定します。次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> <li>- プレーン。メッセージをプレーンテキストで送信します。</li> <li>- HTML。HTML 形式で本文を送信します。HTML の本文にはリストとハイパーリンクを含めることができます。</li> </ul> <p>HTML ビューでは、デフォルトで HTML の基本構造を提供します。HTML を編集すると、デフォルトの構造を復元することはできません。</p>
メッセージ	入力時に電子メールの本文を表示します。
プレビュー	電子メールの本文を、受取人側で表示されるのと同じように表示します。

## タスクインスタンス通知のワークフロー変数

ワークフロー変数を使用して、ヒューマンタスクインスタンスに関する情報を電子メール通知に書き込みます。ユーザーがタスクインスタンスを完了、エスカレート、または再割り当てすると、この変数にタスクインスタンスに関する情報が保存されます。変数の一覧を表示するには、件名または通知メッセージの本文をクリックして、`$+CTRL+SPACE` キーを押します。

通知には、次の変数を表示できます。

`$taskEvent.eventTime`

ワークフローエンジンがユーザーの指示を実行して、タスクインスタンスをエスカレート、再割り当て、または完了したときの時刻。

`$taskEvent.owner`

ワークフローエンジンがタスクをエスカレートまたは完了した時点でのタスクインスタンスの所有者。または、エンジンがタスクインスタンスを再割り当てした後のタスクインスタンスの所有者。

`$taskEvent.status`

エンジンがユーザーの指示を実行して、タスクインスタンスをエスカレート、再割り当て、または完了した後のタスクインスタンスのステータス。ステータスの名前は、READY および IN\_PROGRESS です。

\$taskEvent.taskEventType

エンジンが実行する指示のタイプ。変数の値は、escalate、reassign、complete です。

\$taskEvent.taskId

Analyst ツールが表示するタスクインスタンス ID。

## ヒューマンタスクとワークフローの設定

マッピングタスクとヒューマンタスクを含むワークフローを設定するときは、マッピングタスクをもう 1 つ追加して、ヒューマンタスクで処理したレコードを前のマッピングタスクのレコードと再結合できます。

例えば、次の手順に従って、例外レコードを管理するワークフローを設定できます。

1. ワークフローを作成し、開始イベントと終了イベントを追加します。
2. データセットに重複レコードがないかどうかを調べる *Mapping\_1* というマッピングを作成します。このマッピングに含まれる例外トランスフォーメーションで、品質が正常なレコードを *Good\_Records* という名前のデータベーステーブルに書き込み、例外を *Exceptions* という名前のテーブルに書き込みます。
3. ワークフローにマッピングタスクを追加し、*Mapping\_1* を実行するようにタスクを設定します。
4. ワークフローにヒューマンタスクを追加し、*Exceptions* テーブルのデータをユーザーに割り当てるようにタスクを設定します。
5. *Mapping\_2* というマッピングを作成し、*Exceptions* テーブルを読み取って、ヒューマンタスクでステータスが「承認済み」に更新されたレコードを *Good\_Records* テーブルに書き込むように設定します。
6. ワークフローにマッピングタスクをもう 1 つ追加し、そのマッピングタスクに *Mapping\_2* を追加します。  
選択したユーザーに電子メールメッセージを送信するように通知タスクを設定します。その電子メールメッセージで、マッピングタスクが完了したことを通知します。
7. ワークフローでイベントとタスクを接続します。

ワークフローを実行すると、最初のマッピングで *Exceptions* テーブルが作成され、2 番目のマッピングで *Exceptions* テーブルから *Good\_Records* テーブルに品質が正常なレコードが書き込まれます。

## ヒューマンタスクの設定

ヒューマンタスクは、タスクインスタンスとステップで構成されています。

タスクインスタンスは、データセットを複数のユーザーに分割する際に作成します。ステップは、ユーザーの実行するアクションを定義する際に作成します。さらに、タスクデータがステップに渡されたときのタスクインスタンスを処理するユーザーを指定します。

### ヒューマンタスクのタスクインスタンスの設定

ワークフローのヒューマンタスクで、タスクインスタンスを設定します。

ヒューマンタスクの【プロパティ】ビューを使用してタスクオプションを設定します。[プロパティ] ビューが表示されない場合、【ウィンドウ】 > 【ビューの表示】 > 【プロパティ】を選択します。

1. Developer ツールでワークフローを開き、ヒューマンタスクを追加します。
2. 【全般】タブで、ヒューマンタスクの名前とオプションの説明を確認します。



3. **【パーティシパント】** タブで、タスクインスタンスを管理するビジネス管理者を特定します。
4. **【データソース】** タブで、例外レコードを格納するデータベースのデータベース接続情報を指定します。
5. **【タスク分散】** タブで、ワークフローがデータをタスクインスタンスに割り当てる際に使用するオプションを設定します。
6. **【通知】** タブで、ヒューマンタスクが完了したときに通知するユーザーを指定し、電子メールで送る通知文を作成します。
7. **【入力】** タブで、ヒューマンタスクの入力を定義します。  
入力を、ワークフローのマッピングタスクから *exceptionLoadCount* 変数を読み込むように設定します。
8. オプションとして、**【出力】** タブで、1 つまたは複数の変数に対してタスク情報を書き出すように出力を設定します。  
この出力は、開始時刻、終了時刻、タスクの実行結果を示します。

## タスクステップの設定

ヒューマンタスクにステップを 1 つまたは複数作成します。例外ステップとクラスタステップが追加できます。例外ステップまたはクラスタステップの作業を検証する場合は、オプションとして確認ステップを作成します。

1. ワークフローからヒューマンタスクを開きます。
2. **【全般】** タブで、ステップの名前とオプションの説明を確認します。
3. **【パーティシパント】** タブで、このステップのタスクを処理できるユーザーまたはグループを選択します。  
そのタスクの作業を行うユーザーまたはグループを指定し、タスクの割り当てについての通知を受け取るビジネス管理者を指定します。
4. **【タイムアウト】** タブで、ステップタスクを完了する必要がある期間を設定します。
5. **【通知】** タブで、ステップに関連付けられたタスクインスタンスのステータスが変化したときに通知を受け取るユーザーを指定します。
6. **【設定】** タブで、確認するステップ（例外またはクラスタ）を選択します。**【設定】** タブが表示されるのは、確認タスクのみです。

## タスクインスタンスの作成

タスク分散オプションを使用してヒューマンタスクのインスタンスを作成します。

データセット内の項目数に基づいてタスクインスタンスを作成するオプションや、選択したカラムの値に基づいてタスクインスタンスを作成するオプションを設定できます。

テキストに基づいてタスクインスタンスを作成するオプションを選択した場合は、タスクステップを設定するときに使用する必要があるユーザーリストを定義します。項目数に基づいてタスクインスタンスを作成するオプションを選択した場合は、ユーザーリストは定義しません。

**注:** タスク設定のポリシーを変更すると、前のタスク設定は破棄されます。



## 同じサイズのタスクインスタンスの作成

データセットに含まれるレコードまたはクラスタの数に基づいてタスクインスタンスを作成することができます。また、作成するタスクインスタンスの数を指定することもできます。いずれの場合も、同じサイズのタスクを作成します。

1. ヒューマンタスクを開きます。
2. **【タスク分散】** タブを選択し、タスク分散を有効にします。
3. 項目数に基づいてタスクインスタンスを作成するオプションを選択します。
4. 作成するタスクの数を設定するか、各タスクの行またはクラスタの数を設定します。

## データ値に基づくタスクインスタンスの作成

選択したカラム内に共通の値を持つデータセットがある場合、そのデータセット内にすべてのレコードが含まれるタスクインスタンスを作成できます。

1. ヒューマンタスクを開きます。
2. **【タスク分散】** タブを選択し、タスク分散を有効にします。
3. データ値に基づいてタスクインスタンスを作成するように選択します。  
必要に応じて、タスクインスタンスに含めることができるレコードの最大数を指定します。
4. **【カラム】** メニューでカラム名を選択します。メニューには、[データソース] タブで指定したリソースのカラム名が表示されます。  
ヒューマンタスクにクラスタステップを追加する場合は、クラスタの生成元のマッピングで使用したグループキーカラムを選択します。グループキーのカラムを選択して、タスク分散プロセスによってクラスタ内のレコードが複数のクラスタに分割されないようにします。  
**注:** [カラム] メニューの精度は 65 です。65 文字より長い文字列が含まれている列はメニューに表示されません。
5. ユーザーまたはグループをタスクインスタンスに割り当てるためにワークフローで使用方法を選択します。  
各ユーザーまたはグループはカラムの 1 つの値に関連付けることも、カラム値の範囲に関連付けることもできます。
6. ユーザーまたはグループと、カラムデータ値の間の割り当てを作成します。  
**【タスク分散】** タブで、ユーザーまたはグループと、それらに関連付ける値をグリッドに追加できます。または、ユーザー、グループ、およびカラム値の一覧を格納する外部データベーステーブルに接続できます。
7. グリッドに値を追加するには、次のいずれかのオプションを選択します。
  - **データ値の追加。** 値を入力し、ドメインからユーザーまたはグループを選択します。
  - **参照テーブルからのデータ値の追加。** 参照テーブルからデータ値、およびユーザーまたはグループ名をインポートします。ユーザー名またはグループ名のカラムと、データ値のカラムを含む参照テーブルを作成します。
  - **ローカルファイルからのデータ値の追加。** 区切りファイルからデータ値およびユーザーまたはグループ名をインポートします。
8. 外部データベーステーブルから値を追加するには、データベース接続を参照し、データベーステーブルを選択します。データベースからのリソース。
9. 必要に応じて、指定したタスク分散条件を満たさないデータソース内のレコードを受信するユーザーまたはグループを選択します。

## カラム値ごとのタスク分散に関するルールおよびガイドライン

タスク分散でデータ値とユーザー名またはグループ名を指定する場合は、次のルールとガイドラインに従ってください。

- **【タスク分散】** タブのグリッドに、データ値とユーザー名またはグループ名を入力することができます。または、テーブルまたはファイルからデータ値とユーザー名またはグループ名をインポートできます。
- データ値とユーザー名またはグループ名の各組み合わせを 1 行に入力します。
- 値をユーザーまたはグループに関連付けるには、最初のカラムに 1 つの値を、2 番目のカラムにはユーザーまたはグループを 1 つ入力します。  
数値の範囲をユーザーまたはグループに関連付けるには、最初のカラムに最小値を、2 番目のカラムに最大値を入力します。次に、3 番目のカラムにユーザー名またはグループ名を入力します。  
日付の範囲をユーザーまたはグループに関連付けるには、最初のカラムに最初の日付を、2 番目のカラムに最後の日付を入力します。次に、3 番目のカラムにユーザー名またはグループ名を入力します。
- 4 つ以上のカラムが含まれるテーブルを選択できます。タスクの設定では、単一のデータ値を指定した場合は最初の 2 カラム、値の範囲を指定した場合は最初の 3 カラムが使用されます。タスクの設定では、追加のカラムは無視されます。
- 最小値と最大値は、**【タスク分散】** タブの開始値と終了値に対応します。最初の日付と最後の日付は、開始日と終了日に対応します。
- 各行で開始値が終了値以下である必要があります。開始日は終了日以前である必要があります。
- ワークフローでは、各範囲の両端の値を含めて扱います。例えば、50 - 99 の範囲には、50 と 99 の値が含まれます。設定した範囲は重複できません。
- 指定するユーザーとグループは、ワークフローが実行される Informatica ドメインに存在する必要があります。参照テーブルまたはファイル内のデータ値は、選択したデータソースカラム内の値と一致する必要があります。  
ユーザーまたはグループ名にセキュリティドメイン名を含めます。例えば、*Native|Jefferson* と入力して、ネイティブドメインのユーザー名 Jefferson を指定します。
- データ値に区切り文字が含まれている場合は、引用符をテキスト修飾子として使用してデータ値を囲みます。
- カラム値または値の範囲を複数のユーザーまたはグループに関連付けることができます。ワークフローが実行されると、データ統合サービスでは、値または値の範囲を含むすべてのタスクを、分散リスト内の最初のユーザーまたはグループに割り当てます。
- レコードに分散条件と一致しない値が含まれている場合、データ統合サービスはこのレコードを固有のタスクインスタンスに追加します。データ統合サービスでは、**【タスク分散】** タブで指定したユーザーまたはグループにインスタンスを割り当てます。  
ドメイン内の任意のユーザーまたはグループ（他のタスクを受信するように選択したユーザーまたはグループを含む）を選択できます。

## ステップに対する権限の設定

タスクの簡単な説明をここに入力します（オプション）。

1. ヒューマンタスクを開きます。
2. **【権限】** タブを選択します。  
タブに、表示権限と編集権限のオプションが表示されます。
3. 各権限を設定するには、**【値】** フィールドのメニューを開きます。メニューオプションを使用し、必要な権限を選択します。

- 選択したカラムに表示権限を設定することを選択すると、**【表示列】** ダイアログボックスが開きます。ダイアログボックスオプションを使用し、Analyst ツールで表示または非表示するカラムを選択します。
- 選択したカラムに編集権限を設定することを選択すると、**【編集可能な列】** ダイアログボックスが開きます。ダイアログボックスオプションを使用し、ユーザーが Analyst ツールで編集できるカラムを選択します。

**注:** 1 つ以上の表示列に編集権限を定義します。Analyst ツールユーザーは、表示できないカラムのデータを編集できません。

## 関連項目：

- [「ステップの権限オプションのルールおよびガイドライン」 \(ページ 83\)](#)
- [「権限オプション」 \(ページ 76\)](#)

## ステップの権限オプションのルールおよびガイドライン

デフォルトでは、Analyst ツールユーザーは、すべてのデータを表示し、所有するタスクインスタンスのアクションを実行できます。タスクインスタンスに一連の表示権限と編集権限を指定するには、**【権限】** オプションを使用します。タスクインスタンスを定義するステップで権限を適用します。

権限を使用し、ユーザーのロールまたは専門分野に基づいてデータをユーザーに割り当てることができます。例えば、複数のユーザーがワークフローデータの複数のカラムを編集できるように、一連のステップをヒューマンタスクに定義できます。または、給与や社会保障番号など、1 人以上のユーザーの機密データを非表示にすることができます。

権限を設定する場合、次のルールとガイドラインを検討します。

- ヒューマンタスクの各ステップに異なる権限を定義できます。タスクインスタンスを確認する Analyst ツールユーザーに、以前のタスクインスタンスのデータを処理したユーザーに設定されている別のデータが表示される場合があります。
- 権限は、各タスクインスタンスのレコードまたはクラスターデータに加えて、監査証跡データに適用されます。
- ビジネス管理者がタスクインスタンスを別のユーザーに転送する場合、ステップに設定した権限は、ビジネス管理者が指定したユーザーに適用されます。
- ステップに設定した表示権限は、編集権限よりも優先されます。カラムの編集権限を付与してカラムの表示権限を付与しない場合、Analyst ツールユーザーはカラムデータを編集できません。
- タスクインスタンスで複数の値の検索と置換を行う場合、表示権限によってタスクインスタンスに指定されたデータに操作が限定されます。
- Analyst ツールユーザーは、設定した表示権限または編集権限に関係なく、レコード、クラスター、またはタスクインスタンスのステータスを更新するアクションを実行できます。

## 関連項目：

- [「権限オプション」 \(ページ 76\)](#)

## 第 7 章

# マッピングタスク

この章では、以下の項目について説明します。

- [マッピングタスクの概要, 84 ページ](#)
- [マッピングタスクの全般プロパティ, 85 ページ](#)
- [マッピングタスクの入力, 86 ページ](#)
- [マッピングタスクの出力, 90 ページ](#)
- [マッピングタスクログ, 103 ページ](#)
- [マッピングタスクの詳細プロパティ, 104 ページ](#)
- [タスクで実行するマッピングの変更, 110 ページ](#)
- [クラスタワークフロー内のマッピングタスク, 110 ページ](#)

## マッピングタスクの概要

マッピングタスクは、ワークフローでマッピングを実行するタスクです。

ワークフローにマッピングタスクを追加するときは、タスクで実行する単一のマッピングを選択します。物理データオブジェクトを入力および出力とする任意のマッピングを選択できます。マッピングを変更すると、Model Repository サービスによって、そのマッピングを含むすべてのマッピングタスクに対するそれらの変更による影響が追跡されます。

ワークフローを実行すると、マッピングタスクのパラメータおよび変数に入力データが渡され、その入力データを使用してマッピングタスクでマッピングが実行されます。マッピングタスクでのマッピングの実行が完了すると、出力データがワークフローの変数に渡されます。

マッピングタスクを設定する際、タスクの入力データおよび出力データを指定します。マッピングタスクのランタイムコンテキストを作成するパラメータを設定できます。場所のタスクプロパティおよびマッピングログファイルのファイル名を設定します。最適化レベル、デフォルトのソート順、Java クラスパスなどのその他のプロパティを、マッピングタスクレベルで設定します。

マッピングタスクを設定した後、タスクで実行する別のマッピングを選択できます。

## 同じマッピングを実行する複数のマッピングタスク

同じマッピングを含む複数のマッピングタスクを 1 つのワークフローに含めることができます。

同じマッピングを含む複数のマッピングタスクがワークフローに含まれている場合、マッピングを別々に実行するようにマッピングタスクを設定することができます。また、それぞれ異なるパラメータ値を使用してマッ

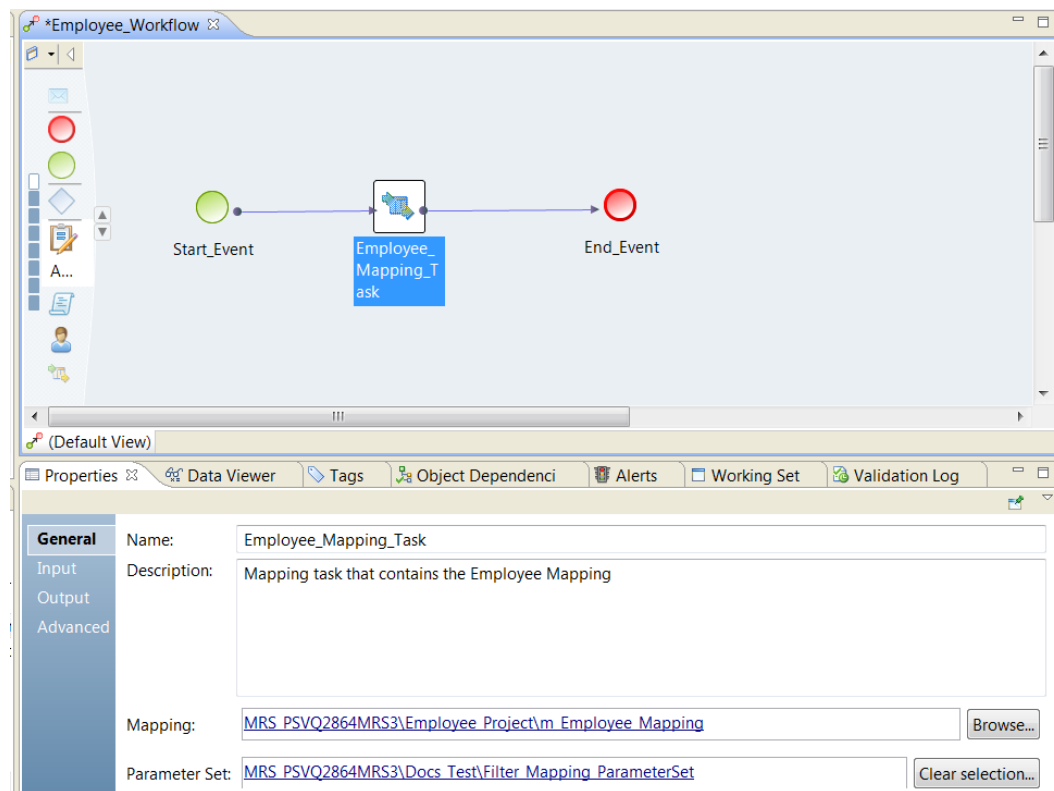
ピングを実行できるように、マッピングタスクのユーザー定義マッピングパラメータにワークフローのパラメータや変数を割り当てることもできます。

例えば、どちらも MappingA を含む 2 つのマッピングタスクを 1 つのワークフローに追加するとします。一方のマッピングタスクは Sales データベースに接続して MappingA を実行するように設定し、もう一方のマッピングタスクは Finance データベースに接続して MappingA を実行するように設定することができます。

## マッピングタスクの全般プロパティ

マッピングタスクのマッピングを変更できます。マッピングタスクにパラメータセットを追加することもできます。

次の図は、マッピングタスクの [全般] ビューを示しています。



[全般] ビューを表示するには、ワークフローでマッピングタスクをクリックします。[全般] ビューをクリックします。

[全般] ビューの次のフィールドを変更できます。

### 名前

マッピングタスクの名前。

### 説明

マッピングタスクの説明。

## マッピング

マッピングタスクに含めるマッピングの名前およびパス。**【参照】**をクリックして、リポジトリで別のマッピングを選択します。

## パラメータセット

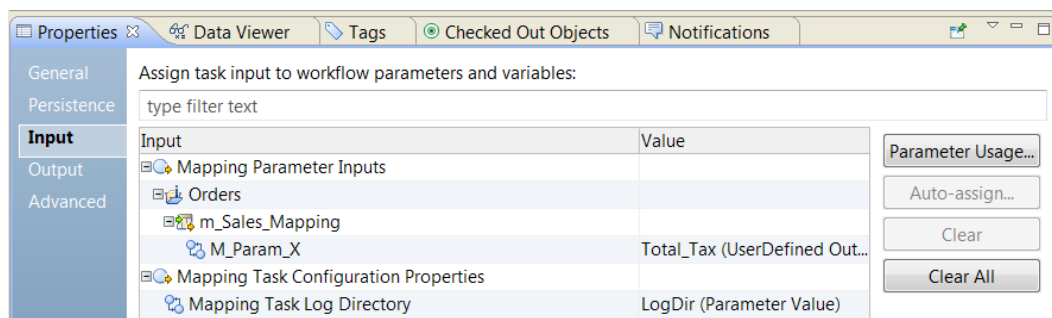
マッピングタスクで使用するパラメータセットの名前およびパス。**【参照】**をクリックして、パラメータセットを検索します。**【選択のクリア】**をクリックして、パラメータセットを削除します。パラメータセットを表示するには、パラメータセットへのリンクをクリックします。

# マッピングタスクの入力

マッピングタスクの入力とは、マッピングタスクに渡すデータです。マッピングタスクは、マッピングパラメータまたはマッピングタスクの設定プロパティから入力データを受け取ることができます。

マッピングタスクの**【入力】**タブで、マッピングタスクの入力に値を割り当てます。

次の図は、マッピングタスクの**【入力】**タブを示しています。



次のタイプのマッピングタスクの入力に、値を割り当てることができます。

## マッピングパラメータの入力

マッピングパラメータとは、マッピングレベルで定義するパラメータです。**【入力】**タブには、マッピングでフィールドに割り当てていないパラメータを含め、マッピングのすべてのパラメータが一覧表示されます。

## マッピングタスクの設定プロパティ

マッピングタスクの設定プロパティとは、**【詳細】**タブで設定するプロパティです。マッピングタスクの設定プロパティにパラメータを割り当てる必要がある場合、プロパティに**【タスク入力に割り当てる】**オプションを選択できます。このオプションを選択すると、**【入力】**タブに設定プロパティが表示され、値を割り当てることができます。

マッピングタスクの入力に値を割り当てるには、パラメータまたは設定プロパティの**【値】**カラムをクリックします。マッピングの出力、ワークフロー変数、およびパラメータのリストが表示されます。リストから出力、変数、またはパラメータを選択するか、新しい値を作成できます。

## 関連項目：

- [「タスク入力へのワークフローパラメータの割り当て」](#) (ページ 40)
- [「タスク入力への変数の割り当て」](#) (ページ 33)

## マッピングパラメータとマッピングタスク

データ統合サービスでは、ワークフロー内のマッピングの実行時にユーザー定義マッピングパラメータの値を使用することができます。ワークフローにパラメータセットまたはパラメータファイルを使用できます。

ワークフローを実行する際、実行時にパラメータセットを指定できます。パラメータセットには、ユーザー定義のパラメータ値が含まれます。ワークフローのデプロイ時には、アプリケーションにパラメータセットを追加する必要があります。ワークフローのデプロイ時に複数のパラメータセットを含めて、実行ごとに異なるパラメータセットでワークフローを実行できます。

マッピングタスクにパラメータセットを追加することもできます。マッピングタスクにパラメータセットを追加すると、パラメータセットによりユーザーが定義したマッピングパラメータ値またはワークフローパラメータ値が上書きされます。

ワークフローの実行時には、ワークフローに対して 1 つのパラメータファイルを指定できます。ユーザー定義マッピングパラメータの値は、パラメータファイルの mapping 要素または mapplet 要素で定義します。ワークフローパラメータの値は、パラメータファイルの workflow 要素で定義します。

パラメータファイルまたはパラメータセットでワークフローを実行すると、データ統合サービスはすべてのユーザー定義のマッピングパラメータ値とワークフローパラメータ値を適用します。

## ワークフローの実行時のマッピングパラメータのオーバーライド

マッピングタスクの【入力】タブでユーザー定義マッピングパラメータをワークフローのパラメータまたは変数に割り当てることで、ワークフローの実行時にユーザー定義マッピングパラメータの値をオーバーライドすることができます。

マッピングで使用するシステムマッピングパラメータをワークフローパラメータに割り当てることはできません。マッピングでユーザー定義マッピングパラメータをフラットファイルディレクトリフィールド、キャッシュファイルディレクトリフィールド、または一時ファイルディレクトリフィールドに割り当てる場合は、ユーザー定義マッピングパラメータをワークフロー文字列パラメータに割り当てることができます。

ユーザー定義マッピングパラメータの値をオーバーライドする状況としては、次のような場合が考えられます。

### ユーザー定義マッピングパラメータの値にランタイムデータを使用する。

ユーザー定義マッピングパラメータの値にワークフローのランタイムデータを使用する場合は、パラメータをワークフロー変数に割り当てます。例えば、ワークフロー文字列変数を使用して、フラットファイル名を定義するマッピング文字列パラメータをオーバーライドするとします。排他ゲートウェイを使用して条件を評価し、割り当てタスク A または割り当てタスク B を実行します。割り当てタスク A ではワークフロー変数を FlatFileA.txt に設定し、割り当てタスク B ではワークフロー変数を FlatFileB.txt に設定します。マッピングタスクの入力で、ワークフロー変数をマッピング文字列パラメータに割り当てます。これにより、マッピングタスクでマッピングを実行するときに、ワークフロー変数に動的に割り当てられる値がソースファイル名として使用されます。

### ワークフローで複数回使用されるユーザー定義マッピングパラメータに異なる値を割り当てる。

1 つのワークフローに同じマッピングを実行するマッピングタスクが複数含まれている場合は、マッピングタスクごとに異なるワークフローパラメータにユーザー定義マッピングパラメータを割り当てます。パラメータファイルで、ワークフローパラメータごとに異なる値を定義します。パラメータファイルを指定してワークフローを実行すると、各マッピングタスクで実行されるマッピングで、それぞれに割り当てられたワークフローパラメータの値が使用されます。

例えば、ワークフローに含まれる 2 つのマッピングタスクの両方に、SourceConnection というユーザー定義マッピングパラメータを使用する同じマッピングが含まれているとします。ワークフローの実行時にそれぞれのマッピングで異なるソース接続を使用する場合は、マッピングタスクごとに異なるワークフロー接続パラメータにマッピングパラメータ SourceConnection を割り当てます。MappingTask1 ではワークフローパラメータ WF\_SourceConnection1 に SourceConnection を割り当て、MappingTask2 では



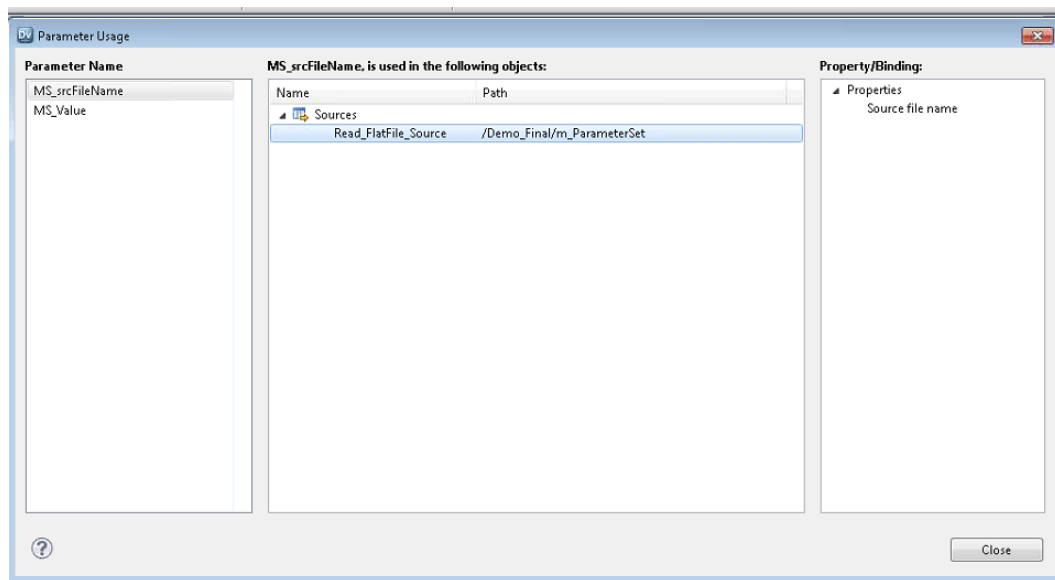
ワークフローパラメータ WF\_SourceConnection2 に SourceConnection を割り当てます。パラメータセットをマッピングタスクに割り当てます。パラメータセットで、ワークフローパラメータ WF\_SourceConnection1 および WF\_SourceConnection2 に異なる値を設定します。

## マッピングタスクでのパラメータの使用

マッピングパラメータを使用するマッピングタスク内のオブジェクトのリストを表示できます。

マッピングパラメータを使用するマッピングタスク内のオブジェクトを表示するには、マッピングタスクの **[入力]** ビューから **[パラメータの使用]** をクリックします。**[パラメータの使用]** ダイアログボックスに、マッピングタスク内のすべてのマッピングパラメータが表示されます。マッピングパラメータを選択して、そのパラメータを使用するオブジェクトを表示します。

次の図に、**[パラメータの使用]** ダイアログボックスを示します。



**[パラメータの使用]** ダイアログボックスは、次のフィールドで構成されます。

### パラメータ名

マッピングでのマッピングパラメータの名前。

### 名前

マッピングパラメータを使用するオブジェクトのタイプおよび名前。

### パス

リポジトリ内のオブジェクトへのパス。フォルダ名とマッピング名が含まれます。

### プロパティ/バインディング

マッピングパラメータにバインドされているプロパティを示します。マッピングパラメータはユーザー定義のパラメータまたは特定のフィールドにバインドされることがあります。マッピングパラメータがフィールドまたはユーザー定義のパラメータにバインドされている場合、マッピングでは他の値ではなくマッピングパラメータ値が使用されます。



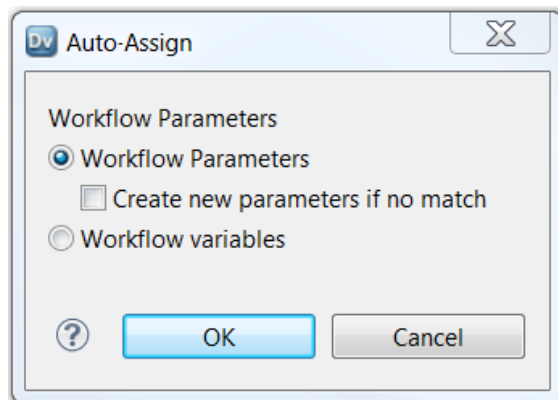
## マッピング入力パラメータ値の自動入力

マッピングタスクの【入力】タブでマッピングパラメータをワークフローパラメータまたはワークフロー変数に自動的に割り当てることができます。

1回の操作でワークフローパラメータまたはワークフロー変数を生成してマッピングパラメータにバインドするには、【自動割り当て】オプションを選択します。マッピングパラメータをワークフローパラメータまたは以前に作成したワークフロー変数に割り当てすることもできます。

マッピングパラメータを自動割り当てするには、【入力】カラムでマッピングパラメータを選択して【自動割り当て】をクリックします。

次の図は、【自動割り当て】ダイアログボックスを示しています。



以下のオプションから選択して、マッピングパラメータをワークフローパラメータまたはワークフロー変数に割り当てます。

### ワークフローパラメータ

マッピングパラメータをワークフローパラメータに割り当てます。マッピングパラメータ名と同じ名前のワークフローパラメータ名が存在している必要があります。Developer tool はパラメータのタイプによってパラメータを照合しません。ワークフローにマッピングパラメータを割り当てるパラメータがない場合、一致が見つからなければ新しいパラメータを作成することもできます。

### 一致がない場合は新しいパラメータを作成する

選択したマッピングパラメータと同じ名前のワークフローパラメータを生成します。ワークフローパラメータは文字列タイプまたは接続タイプです。文字列タイプまたは接続タイプでないマッピングパラメータからワークフローパラメータを作成すると、Developer tool は文字列ワークフローパラメータを作成します。Developer tool は、文字列ワークフローパラメータをマッピングパラメータにバインドします。

### ワークフロー変数

マッピングパラメータをワークフロー変数に割り当てます。ワークフロー変数が存在し、ワークフロー変数名はマッピングパラメータ名と同じである必要があります。【自動割り当て】オプションを使用してワークフロー変数を作成することはできません。

## 割り当てのクリア

マッピング入力の値の割り当てをクリアすることができます。値の割り当てとは、マッピング入力のワークフローパラメータ、ワークフロー変数、またはシステムパラメータへの割り当てです。

1つのマッピング入力の値の割り当てをクリアするには、【入力】カラムでマッピングパラメータを選択します。【クリア】をクリックします。値の割り当てを削除することを確認します。Developer tool により、選択したマッピング入力の値カラムがクリアされます。

マッピング入力の割り当てをすべて削除するには、**[すべてクリア]** をクリックします。マッピング入力の値の割り当てをすべて削除することを確認します。Developer tool により、すべてのマッピング入力の値カラムがクリアされます。

## マッピングタスクの出力

マッピングタスクのマッピング出力は、マッピングタスクの実行に関する情報を提供する値です。

マッピング出力をワークフロー変数にバインドして、別のワークフロータスクで出力を使用できます。出力の値を保持して、マッピングタスクを次回実行するときに、その値を入力パラメータとして使用することもできます。

マッピングタスクは、次のタイプの出力を返します。

### 全般出力

開始時間、終了時間、およびタスクが成功したかどうか。

### システム定義の出力

マッピングで処理されたエラー行、ソース行、ターゲット行の数。

### ユーザー定義の出力

マッピングで各行からフィールドまたは式を集計するように定義したマッピング出力。ユーザー定義の永続的な出力とは、データ統合サービスが前のワークフロー実行からリポジトリに保存した値です。現在の値とは、現在のワークフロー実行からのマッピング出力の値です。

マッピング出力に関する次のタスクを実行します。

#### リポジトリでマッピング出力を保持します。

リポジトリ内にマッピング出力値を保持するようにマッピングタスクを設定するか、保持されたマッピング出力値をマッピングタスクの入力に割り当てることができます。

例えば、マッピングで生成された最後のシーケンス番号を返すようにマッピングタスクを設定できます。シーケンス番号を格納するマッピング出力をリポジトリで保持できます。マッピングタスクを実行したときに、最後のシーケンス番号を開始シーケンス番号として使用できます。

#### 出力をワークフロー変数にバインドします。

現在のマッピングタスク実行のマッピング出力を、ワークフロー変数にバインドできます。出力をワークフロー変数にバインドすると、ワークフローは出力の値をワークフロー内の他のタスクに渡します。以前のマッピングタスク実行で保持されたマッピング出力を、現在の実行のワークフロー変数にバインドすることもできます。

### 関連項目：

- [「タスク出力」 \(ページ 28\)](#)
- [「タスク出力を使用した値の割り当て」 \(ページ 31\)](#)

## システム定義のマッピング出力

マッピングまたはトランスフォーメーションでは、システム定義の出力を返すことができます。ワークフローの後続の実行またはワークフローの次のタスクで、マッピング出力を使用できます。

マッピング出力をリポジトリに保持するか、マッピング出力をワークフロー変数にバインドすることができます。マッピング出力をワークフローの後続の実行で使用するか、ワークフロー内の別のタスク（通知タスクや排他ゲートウェイタスクなど）に値を渡すことができます。

例えば、あるトランスフォーメーションで、システム定義の出力である numberOfErrorRows を返すことができます。numberOfErrorRows には、無効なレコードの数が含まれます。この情報を確認して、プロセスを監視したり、マッピングで処理するデータの品質を把握できます。

次の表は、マッピングの例外トランスフォーメーションで生成される出力データを示しています。

トランスフォーメーション	出力データ	データ型	説明
例外	exceptionLoadCount	Integer	<p>未解決のデータ品質の問題を含み手動での確認を必要とするレコードの数このレコードには、エラーや重複する情報が含まれている可能性があります。</p> <p>Developer tool は、マッピングに複数の例外トランスフォーメーションが含まれている場合でも、exceptionLoadCount 出力を 1 つだけ表示します。マッピングに単一の例外トランスフォーメーションがある場合は、exceptionLoadCount 出力をワークフロー変数に割り当てます。それ以外の場合は、予期せぬ動作が発生する場合があります。</p>

## マッピングの保持出力

マッピング出力を保持してリポジトリに出力値を保存し、それ以降同じマッピングタスクを実行するときにその値を使用できます。以前のマッピングタスク実行で保持されたマッピング出力を、現在のマッピングタスク実行のワークフロー変数に割り当てることもできます。

マッピングタスクの【プロパティ】ビューの【持続】タブをクリックすると、Developer tool にマッピングのすべてのマッピング出力が表示されます。マッピング出力を保持するには、マッピング出力の【持続】を有効にして、保持された値を返すために実行する集計のタイプを選択します。

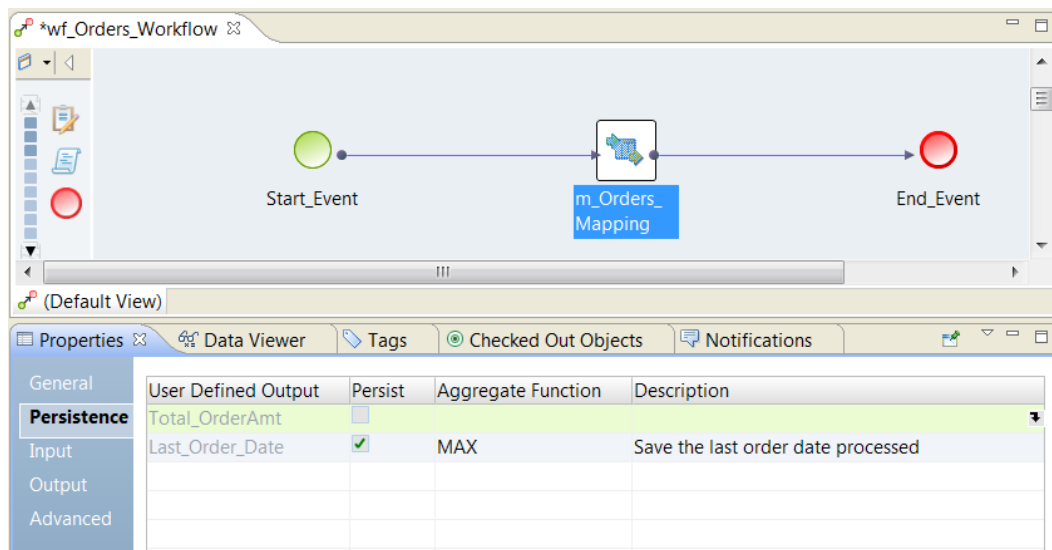
データ統合サービスがモデルリポジトリでマッピング出力を保持する場合、データ統合サービスはマッピングタスク名をキーとしてマッピング出力を保存します。例えば、ワークフローに 4 つのマッピングタスクが含まれ、それぞれが同じマッピングを実行している場合、データ統合サービスはモデルリポジトリ内に 4 つの出力を保存します。

マッピング出力を保持すると、マッピングレベルで定義した集計関数とは異なる集計関数を保持された値について設定できます。データ統合サービスは複数のマッピング出力値を生成します。例えば、OrderDate マッピング出力には MIN OrderDate が含まれていることがあります。保持された OrderDate マッピング出力には、MAX OrderDate が含まれていることがあります。

マッピングタスクのマッピング出力は、マッピングタスクが次回実行されるときに入力パラメータにバインドできます。入力マッピングパラメータにバインドするには、マッピング出力をマッピングタスク内で保持する必要があります。

バインディングの 1 つのユースケースは、マッピングが処理した最後の注文日を保持する場合です。マッピングタスクが次回実行されるとき、マッピングの入力パラメータは最後に処理された日付になります。マッピングでは、最後に処理された注文日より後の注文日を持つ行を含めるように、パラメータソース行をフィルタ処理できます。

次の図は、マッピングタスクの【プロパティ】ビューの【持続】タブを示しています。



【持続】 タブには以下のフィールドがあります。

#### ユーザー定義の出力

マッピングが返すマッピング出力の名前。

#### 持続

データ統合サービスがマッピング出力をリポジトリに保持するようにします。

#### 集計関数

保持するマッピング出力で実行される集計のタイプ。MIN、MAX、または SUM を選択します。デフォルトは、マッピングプロパティに定義したマッピング出力の値です。保持されたマッピング出力の集計関数のタイプは変更できます。ワークフロー変数に渡すマッピング出力値とは異なる値をリポジトリに保持できます。

#### 説明

リポジトリに保持するマッピング出力について説明します。

### 保持された出力の管理

リポジトリ内の保持されたマッピング出力を一覧表示、更新、およびリセットできます。

保持されているマッピングタスク値に対して、次の `infacmd` コマンドを実行できます。

#### `listMappingPersistedOutputs`

ワークフロー内のマッピングタスクインスタンスについて、保持されたマッピング出力およびそれらの値を一覧表示します。

#### `setMappingPersistedOutputs`

ワークフロー内の特定のマッピングタスクインスタンスについて、保持されたマッピング出力を更新またはリセットします。値をリセットすると、保持された値がリポジトリから削除されます。マッピング出力を設定するには、コマンドラインで、マッピング出力の名前と値のペアをスペースで区切って入力します。マッピング出力をリセットするには、マッピング出力のスペース区切りのリストとともに `-reset` オプションを使用します。

`infacmd` の詳細については、『*Informatica コマンドリファレンス*』を参照してください。

## 保持されたマッピング出力およびデプロイメント

ワークフローを再デプロイするか、マッピング出力を変更すると、保持されたマッピング出力の状態に影響することがあります。

保持されたマッピング出力のルールおよびガイドラインは、次のとおりです。

- ワークフローをアプリケーションとして初めてデプロイする際に、マッピングタスクに保持されたマッピング出力がある場合は、追加のタスクを実行する必要はありません。
- アプリケーションを再デプロイする場合は、状態情報の保持または破棄を選択できます。状態情報を保持することを選択した場合は、アプリケーションを再デプロイするときに、マッピング出力値はリポジトリで変更されません。それ以外の場合、マッピング出力の状態は持続から削除されます。
- マッピングまたはワークフローをバックアップおよびリストアするときに、マッピング出力の状態はバックアップされません。
- マッピング出力を名前変更または再作成した場合は、以前のワークフローの実行で保持されたマッピング出力値を使用できません。

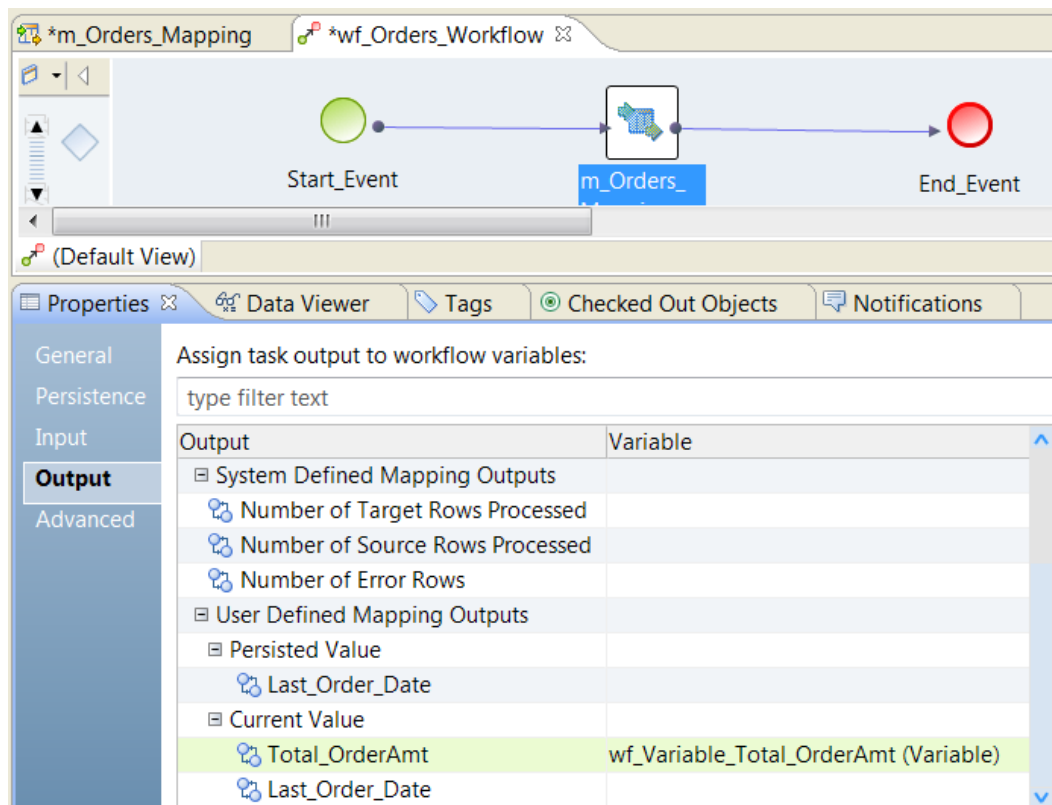
## マッピング出力のワークフロー変数へのバインド

マッピング出力をワークフロー変数にバインドして、値をワークフローのその他のタスクに渡します。

データ統合サービスでマッピング出力を評価して、次に実行するオブジェクトを判断する必要がある場合、マッピング出力をワークフロー変数にバインドすることをお勧めします。または、データ統合サービスが次のタスクのフィールドでマッピング出力値を使用するようにできます。

マッピングタスクの [プロパティ] ビューの **【出力】** タブで、ワークフロー変数に割り当てるマッピングタスクの出力値を設定します。これにより、マッピングタスクの完了時にマッピングタスクの出力値がワークフロー変数にコピーされます。

次の図に、マッピングタスクの **【出力】** タブのマッピング出力をいくつか示します。



【出力】 カラムには、次のタイプのマッピング出力が含まれています。

#### システム定義のマッピング出力

トランスフォーマーがマッピングに返す組み込みのマッピング出力。システム定義マッピング出力には、マッピングが処理したソース行数、ターゲット行数、およびエラー行数が含まれています。

#### ユーザー定義マッピング出力

保持されたマッピング出力値と現在のマッピング出力値。出力値をワークフローの値にバインドできます。

##### 保持された値

以前のワークフローの実行のユーザー定義マッピング出力値。保持された値は、最後のマッピングタスク実行でリポジトリに保持された値です。保持された値は、現在のマッピングで集計された値ではありません。

##### 現在の値

現在のマッピングタスクのユーザー定義マッピング出力値。

ワークフロー変数をマッピングタスクの出力に割り当てるには、[変数] カラムをクリックします。出力を割り当てるワークフロー変数を選択します。例の図では、現在のワークフロー実行からの Total\_OrderAmt マッピング出力がワークフロー変数 wf\_Variable\_Total\_orderAmt にバインドされています。

タスクが失敗した場合は、タスク固有の出力値ではなく、タスクの全般出力の値がワークフロー変数にコピーされます。タスクが強制終了された場合には、タスクの出力値はワークフロー変数にコピーされません。

**注:** マッピング出力をワークフロー変数に割り当てたが、マッピングが行を処理しない場合、出力は NULL になります。マッピングタスクによってワークフロー変数の値は変更されることはありません。変数は、マッピングタスクが実行される前と同じ値を保持します。

#### 例

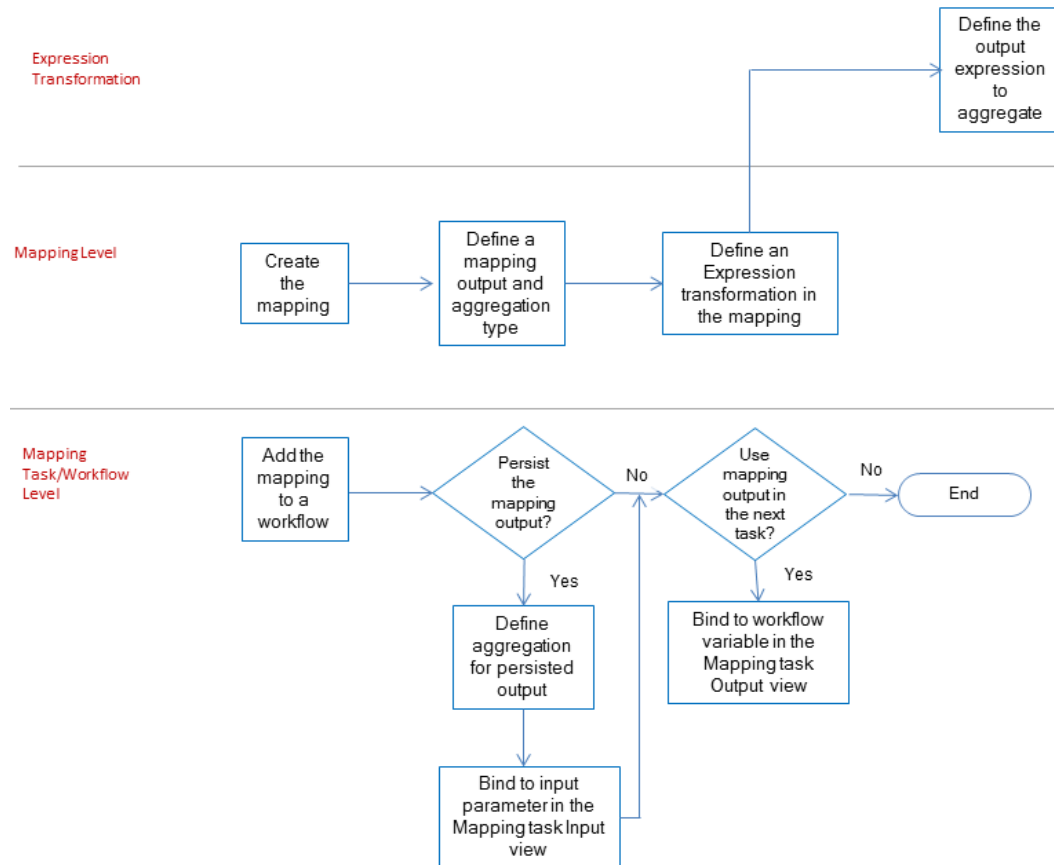
マッピングタスクには、マッピングで処理に失敗した行数をキャプチャする、エラー行数の出力値がいくつか含まれます。このデータを以降のワークフローで使用するには、MappingErrorRows という名前のワークフロ

ー変数にエラー行数の出力を割り当て、その MappingErrorRows ワークフロー変数を排他ゲートウェイからの出力シーケンスフローの式に追加できます。これにより、MappingErrorRows の値が 0 よりも大きい場合と値が 0 の場合で、異なるブランチが選択されるようになります。

## マッピング出力の設定方法

マッピング出力を設定する場合は、マッピングレベルでマッピング出力を定義し、トランスフォーメーションレベルで集計する式を設定して、マッピングタスクレベルで結果を保持します。

次の図は、マッピング出力を設定するプロセスを示しています。



マッピング出力を設定するには、以下の手順を実行します。

1. マッピングを作成します。
2. マッピングの **【出力】** ビューで、マッピング出力名および集計のタイプを定義します。
3. マッピングに式トランスフォーメーションを追加して、式の **【マッピング出力】** ビューでマッピング出力式を設定します。
4. マッピングタスクを作成するには、マッピングをワークフローに追加します。
5. マッピングタスクの **【持続】** ビューでマッピング出力を保持して、保持された値の集計関数タイプを設定します。
6. 保持されたマッピング出力を、マッピングタスクの入力パラメータに割り当てます。
7. マッピング出力を別のワークフロータスクで使用する場合は、マッピング出力をワークフロー変数に割り当てます。

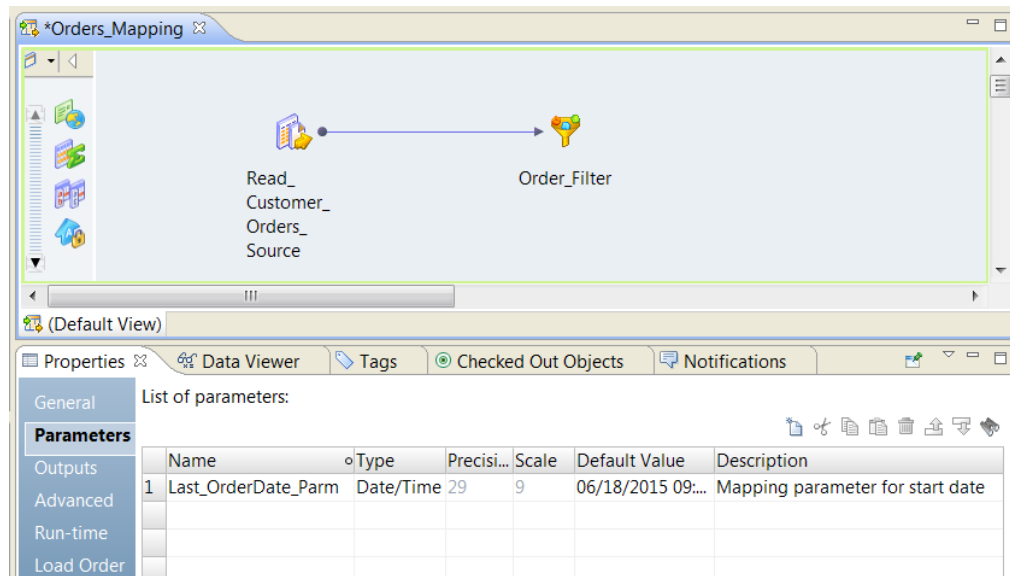
## マッピングの作成

再利用可能なフィルタトランスフォーメーションを含むマッピングを作成します。フィルタトランスフォーメーションは、特定の日付より前の注文日を持つ行をフィルタ処理します。フィルタ式には、Last\_Order\_Date\_Parm というパラメータが含まれています。

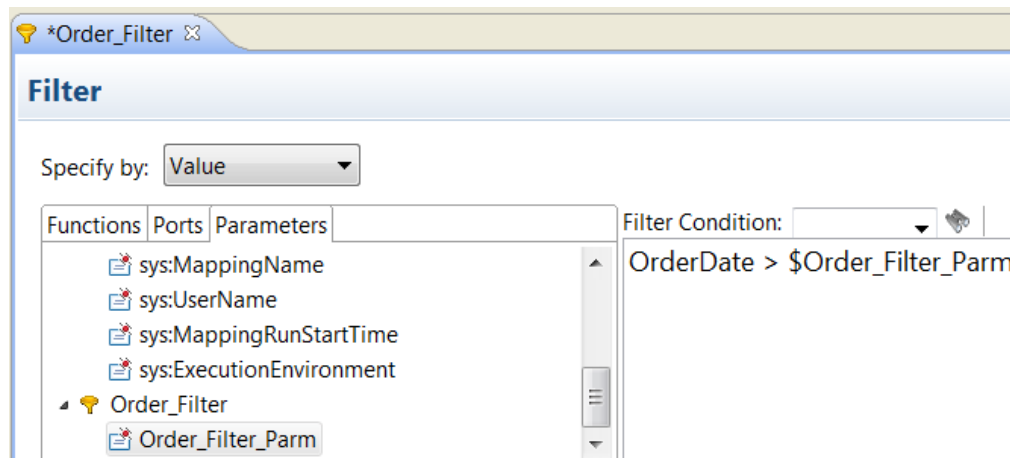
1. Customer\_Order ファイルの注文データを処理するマッピングを作成します。
2. マッピングの [プロパティ] ビューで、[パラメータ] タブをクリックします。
3. Last\_Order\_Date\_Parm という日付/時刻マッピングパラメータを追加します。

開始パラメータのデフォルト日付を入力します。

次の図は、マッピングパラメータを示しています。



4. 再利用可能なフィルタトランスフォーメーションを作成して、Customer\_Order の行をフィルタ処理します。
5. Order\_Filter というフィルタトランスフォーメーションにパラメータを定義します。  
開始パラメータのデフォルト日付を入力します。
6. パラメータより後の注文日を見つけるためのフィルタ式を追加します。

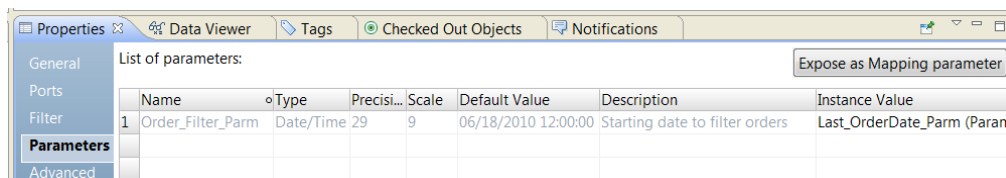


7. フィルタトランスフォーメーションをマッピングに追加します。



8. フィルタトランスフォーメーションをクリックして、トランスフォーメーションの【プロパティ】ビューを表示します。
9. 【パラメータ】タブをクリックします。
10. Order\_Filter\_Parm トランスフォーメーションパラメータを Last\_Order\_Date マッピングパラメータにバインドするには、Order\_Filter\_Parm の【インスタンス値】カラムをクリックします。
11. Last\_Order\_Date を選択します。

次の図は、マッピングパラメータがトランスフォーメーションパラメータにバインドされていることを示しています。



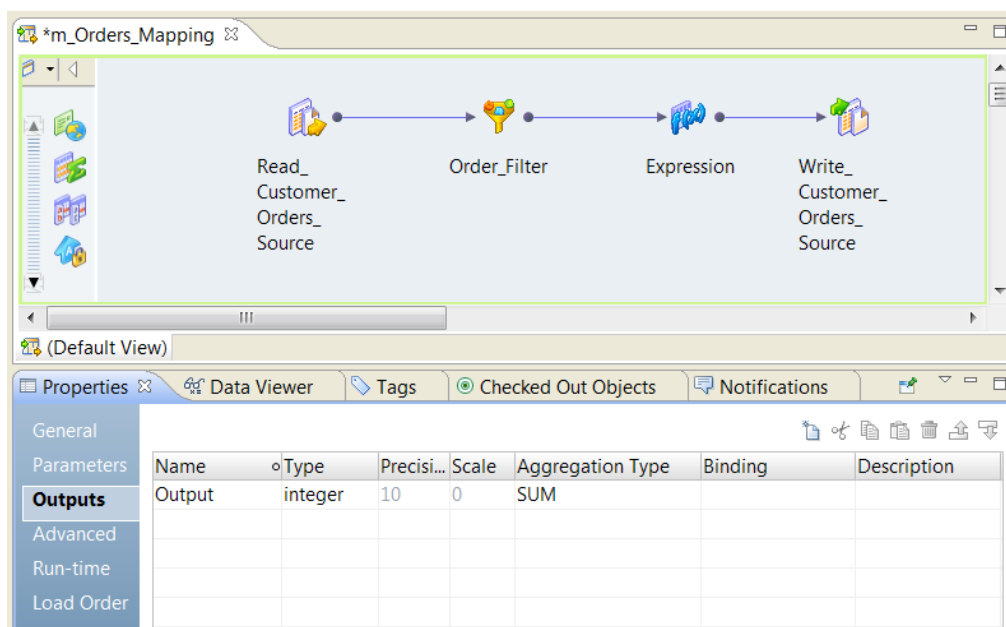
## マッピング出力の定義

マッピングを作成して、マッピングの【プロパティ】でマッピング出力を定義します。各マッピング出力の定義では、実行する集計のタイプ、および結果のデータ型を記述します。

1. マッピングを作成したら、エディタをクリックして、マッピングの【プロパティ】にアクセスします。
2. 【出力】ビューをクリックします。
3. 【新規】をクリックして、マッピング出力を作成します。

Developer tool がマッピング出力を作成して、デフォルトのフィールドの値を設定します。

次の図は、【出力】ビューのマッピング出力のデフォルト値を示しています。



4. マッピング出力を識別する名前を変更します。
5. 数値または日付のマッピング出力タイプを選択します。精度およびスケールを入力します。
6. マッピング出力の集計タイプを選択します。

出力式を集計するか、マッピングが処理した最小式値または最大式値を見つけることができます。デフォルトは SUM です。

7. **【ファイル】 > 【保存】** をクリックして、マッピング出力を保存します。

式トランスフォーメーションにマッピング出力式を作成する前に、マッピング出力を保存する必要があります。

次の図は、decimal フィールドの合計、および最大日付値を含むマッピング出力が含まれるマッピング出力を示しています。

The screenshot shows the 'm\_Orders\_Mapping' tool interface. The top pane displays a workflow with four components: 'Read\_Customer\_Orders\_Source', 'Order\_Filter', 'Expression', and 'Write\_Customer\_Orders\_Source'. The bottom pane shows the 'Outputs' tab with a table of output fields.

Name	Type	Precisi...	Scale	Aggregation Type	Binding	Description
Total_OrderAmt	decimal	10	2	SUM		Add total order amounts
Last_Order_Date	date/time	29	9	MAX		Latest date processed

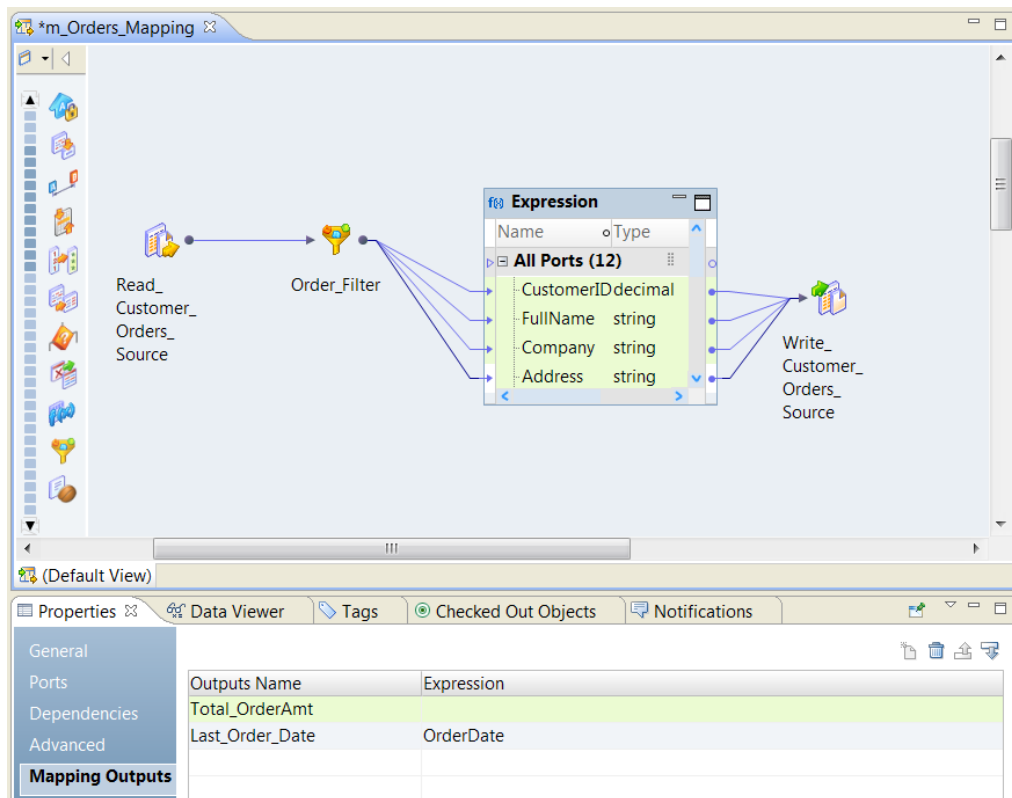
## マッピング出力式の設定

式トランスフォーメーションで、マッピングが処理する各行を集計する式を設定します。

1. 式トランスフォーメーションをマッピングに追加します。  
トランスフォーメーションを配置する場所を決定する前に、マッピングロジックを検討します。マッピング出力には、式トランスフォーメーションが受け取る行の集計が含まれています。
2. 式トランスフォーメーションで、**【マッピング出力】** ビューをクリックします。
3. **【新規】** をクリックして、マッピング出力式を追加します。

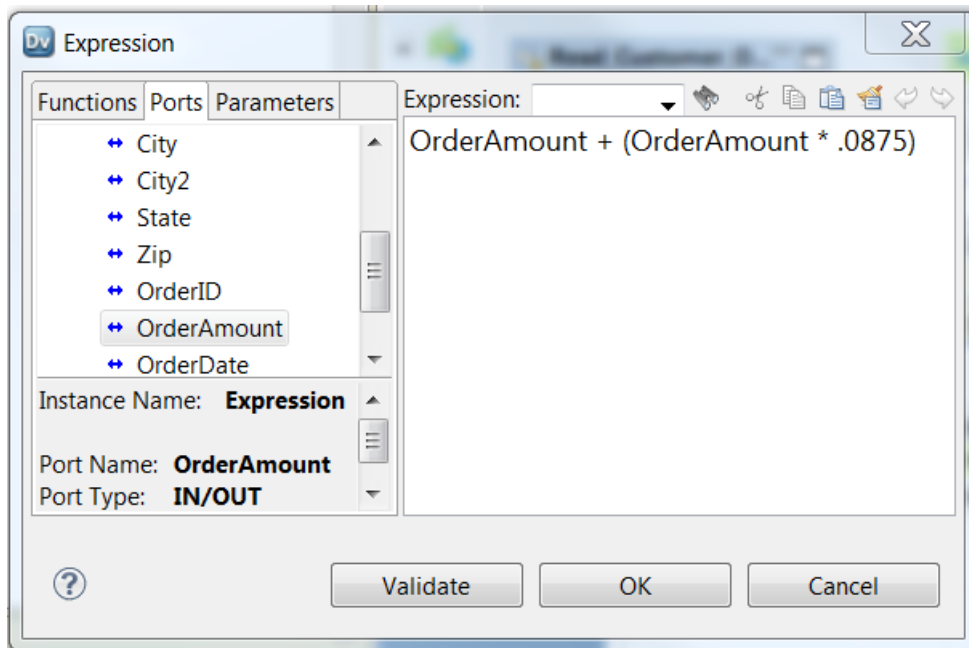
Developer tool が、マッピングレベルで作成したいいずれかのマッピング出力と一致する出力名でマッピング出力を作成します。マッピングの**【プロパティ】** に複数のマッピング出力がある場合は、使用する適切なマッピング出力名を選択します。

次の図は、式トランスフォーメーションの**【マッピング出力】** ビューを示しています。



4. **【式】** カラムをクリックして、式エディタで式を入力します。

式には、ポート名のみを含めるか、関数、ポート、およびパラメータを含めることができます。次の図は、式エディタで Total\_OrderAmt を計算する式を示しています。



5. **【検証】** をクリックして式が有効であることを確認します。

6. **【OK】** をクリックして式を保存します。

式がマッピング出力の【式】 カラムに表示されます。

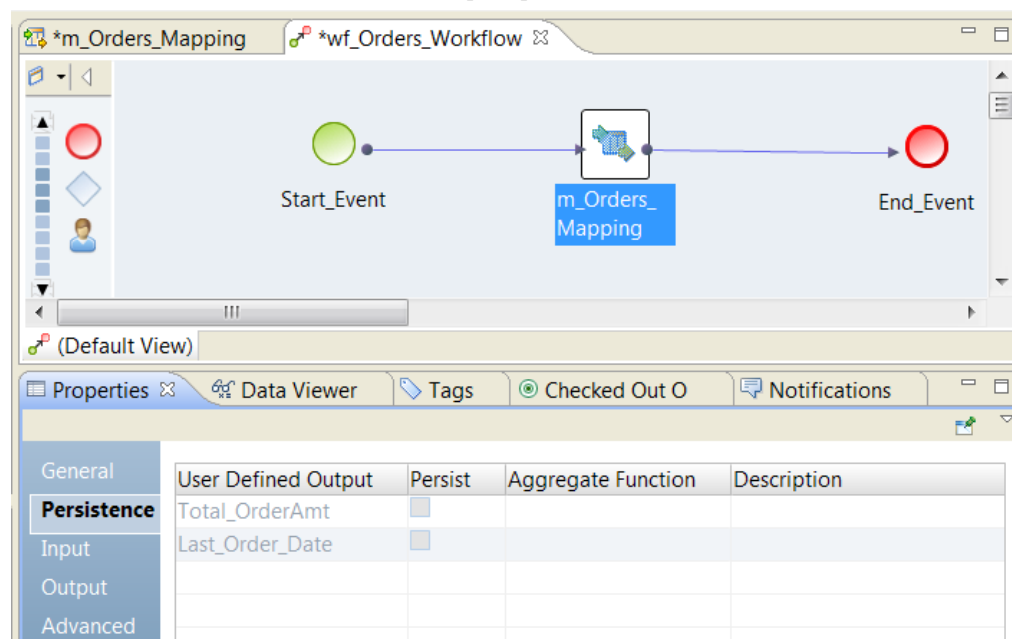
7. 【ファイル】 > 【保存】 をクリックして、式トランスフォーメーションを保存します。

## マッピング出力の保持

マッピングをワークフローに追加したら、マッピングタスクのマッピング出力を保持できます。保持されたマッピング出力を、マッピングタスクが次回実行されるときの入力として使用できます。

1. マッピングをワークフローに追加して、マッピングタスクを作成します。
2. ワークフローの [マッピングタスク] アイコンをクリックして、マッピングタスクの【プロパティ】を表示します。
3. 【持続】 ビューをクリックします。

ユーザー定義のマッピング出力のリストが【持続】 ビューに表示されます。

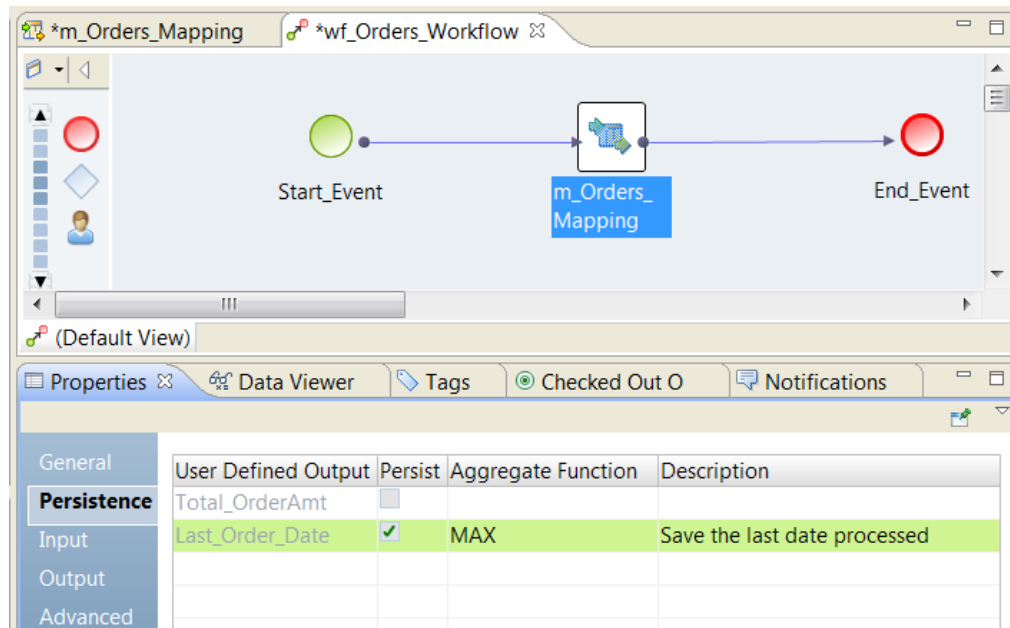


The screenshot shows a workflow editor with a workflow named '\*wf\_Orders\_Workflow'. It contains a mapping task 'm\_Orders\_Mapping' connected between a 'Start\_Event' and an 'End\_Event'. Below the workflow, the 'Properties' panel is open, showing the 'Persistence' tab. The 'Persistence' tab displays a table with the following data:

User Defined Output	Persist	Aggregate Function	Description
Total_OrderAmt	<input type="checkbox"/>		
Last_Order_Date	<input type="checkbox"/>		

4. 【持続】 を有効にして、マッピングタスクが実行された後にマッピング出力を保存します。

- 必要に応じて、集計タイプを変更して説明を入力します。  
次の図は、マッピングタスクの「持続」ビューを示しています。



Last\_Order\_Date マッピング出力が保持されています。集計関数は MAX であるため、データ統合サービスは最大注文日の値をリポジトリに保存します。

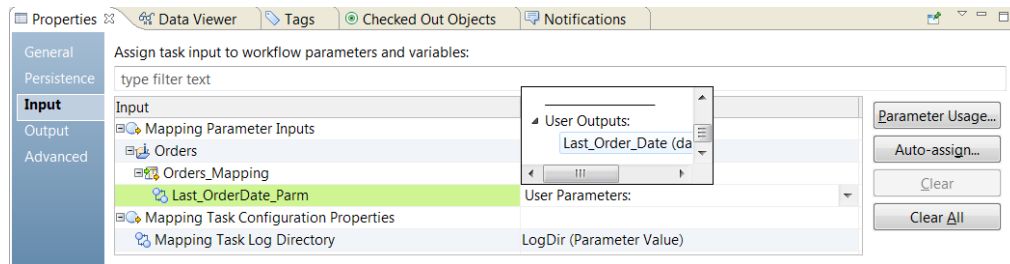
## マッピングタスクの入力への保持された出力の割り当て

マッピングタスクの保持されたマッピング出力は、ワークフローが次回実行される時に同じマッピングタスクの入力パラメータにバインドできます。

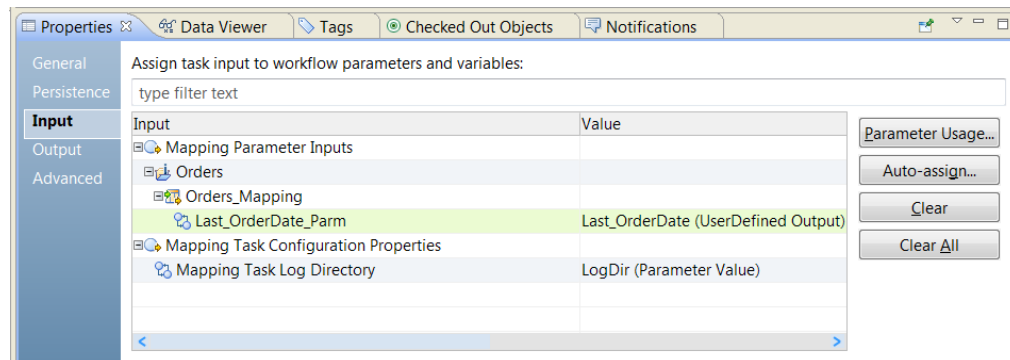
マッピングタスクの保持された最後の注文日の値を、同じマッピングタスクの入力パラメータとして割り当てます。Last\_OrderDate\_Parm パラメータを使用して処理する注文を選択するように、フィルタトランスフォーメーションを設定します。入力行を選択するフィルタ式は、Order\_Date > Last\_OrderDate\_Parm です。

- ワークフローの「マッピングタスク」アイコンをクリックして、マッピングタスクの「プロパティ」ビューを表示します。  
マッピングタスクの入力パラメータのリスト、およびパラメータ化されたマッピングタスク設定プロパティのリストが表示されます。マッピングには、マッピング出力の割り当て先のマッピングパラメータが含まれる必要があります。
- マッピング出力のバインド先となるマッピング入力パラメータを見つけます。「値」カラムをダブルクリックして、選択矢印を表示します。
- 選択矢印をクリックして、入力パラメータに割り当てることができるパラメータおよび変数のリストを表示します。
- リストの「ユーザー出力」セクションをスクロールして、使用する保持されたマッピング出力を選択します。

次の図は、マッピングタスクの「入力」ビューの Last\_OrderDate\_Parm マッピングパラメータを示しています。



5. マッピング出力を選択して、パラメータに割り当てます。  
入力パラメータの値カラムにマッピング出力名が表示されます。



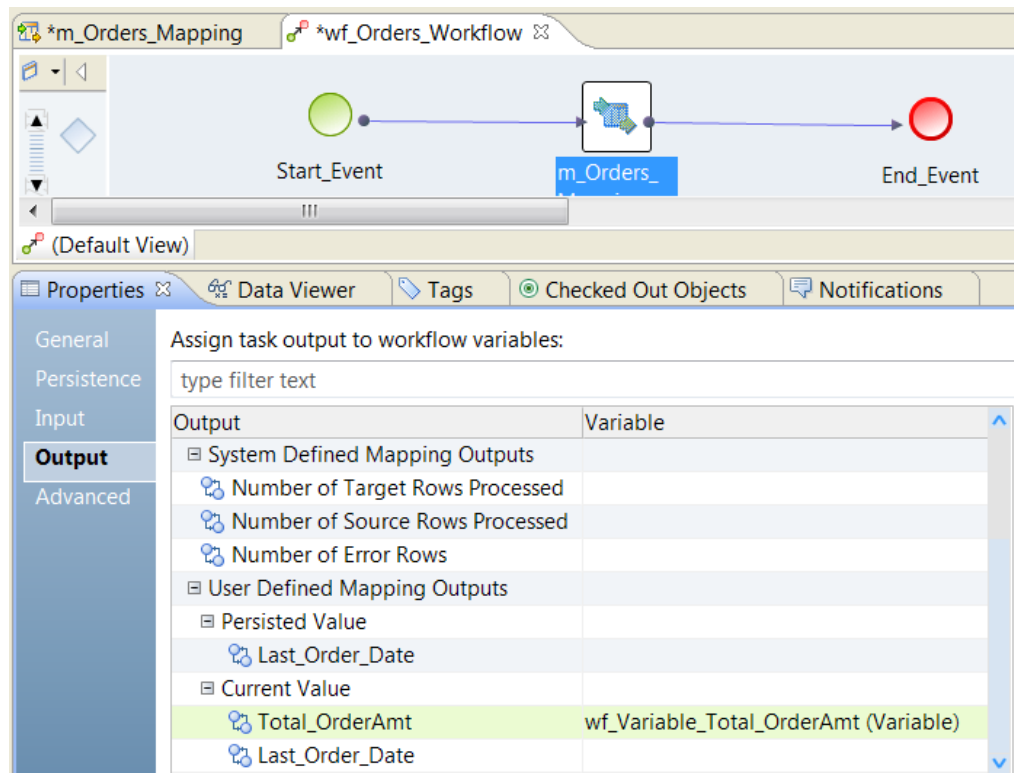
6. **【ファイル】 > 【保存】** をクリックして、マッピングタスクを保存します。  
Last\_OrderDate\_Parm が、リポジトリの保持された注文日の値にバインドされます。

## マッピング出力のワークフロー変数へのバインド

マッピング出力をワークフロー変数にバインドして、値をワークフローの他のタスクに渡すことができます。

マッピング出力値を別のタスクに渡すには、マッピングタスクの**【出力】** ビューでマッピング出力をワークフロー変数にバインドします。現在のマッピングタスクのマッピング出力、または以前のマッピングタスク実行の保持されたマッピング出力をバインドできます。

1. マッピング出力を含むマッピングをワークフローに追加します。
2. ワークフローの**【マッピングタスク】** アイコンをクリックして、マッピングタスクの**【プロパティ】** を表示します。
3. マッピングタスクの**【プロパティ】** で、**【出力】** ビューをクリックします。  
マッピングタスクの**【出力】** ビューに、タスクからワークフロー変数に渡すことができるデータが表示されます。
4. 変数にバインドするマッピング出力を見つけます。
5. **【変数】** カラムをダブルクリックして選択矢印にアクセスし、ワークフロー変数のリストを表示します。  
次の図は、マッピングタスクの**【出力】** ビューで、Total\_OrderAmt マッピング出力を wf\_Variable\_Total\_OrderAmt ワークフロー変数にバインドする場所を示しています。



6. ワークフロー変数を作成するには、【値】カラムのワークフロー変数のリストから【新規変数】オプションをクリックします。

変数名、タイプ、およびデフォルト値を入力します。

## マッピングタスクログ

マッピングタスクログを表示して、マッピングタスクの問題をトラブルシューティングしたり、マッピング実行についての情報を確認できます。

データ統合サービスは、マッピングタスク実行ごとに新しいログファイルを書き込みます。ログファイルには、マッピングタスクのイベントについての情報が含まれます。ログイベントは、タイムスタンプ、スレッド識別子、重要度コード、およびログメッセージを含むテキストの行です。メッセージには、一般情報またはエラーメッセージを含めることができます。

次のテキストは、マッピングタスクログメッセージの形式を示しています。

```
2015-02-20 12:49:24 <DTMLLoggerThread_2> INFO: READER_1_1_1,   DBG_21430,   Reading data from input source
file [C:\Source\Logging_Source_1.txt]
2015-02-20 12:49:24 <DTMLLoggerThread_2> INFO: READER_1_1_1,   BLKR_16019,   Read [200] rows, read [0] error
rows for source table [read_src2] instance name [read_src2]
2015-02-20 12:49:24 <DTMLLoggerThread_2> INFO: LKPDP_2,   TE_7212,   Increasing [Data Cache] size for
transformation [Rel_Lookup] from [59652322] to [59654144].
2015-02-20 12:49:24 <DTMLLoggerThread_2> INFO: READER_1_1_1,   BLKR_16008,   Reader run completed.
2015-02-20 12:49:24 <DTMLLoggerThread_2> INFO: WRITER_1_*_1,   WRT_8167,   Start loading table
[Router_Target_Default] at: Fri Feb 20 12:49:24 2015
```

トレースレベルを詳細 - データに設定すると、マッピングタスクログにはマッピング実行のパラメータおよびパラメータ値が表示されます。

次のテキストは、パラメータ値を含むマッピングタスクログメッセージを示しています。

```
Integration Service will use override value [C:\Source] for parameter [ff_SrcDir] in transformation [map_AllTx\read_src1].
Integration Service will use override value [8] for parameter [exp_Int] in transformation [map_AllTx\Expression].
Integration Service will use override value [Mapping_New] for parameter [exp_String] in transformation [map_AllTx\Expression].
Integration Service will use override value [C:\Source] for parameter [ldo_SrcDir] in mapping \maplet [map_AllTx\DO_Lookup\DO_FF_REL_SRC_Read_Mapping].
```

Hadoop クラスタ上の Spark エンジンでマッピングを実行した後、マッピングの実行に使用されたクラスタノードの合計数を、マッピングタスクログで確認できます。Blaze エンジンでは、グリッドマネージャによって使用された正常なクラスタノードの数を、マッピングタスクログで確認できます。

## マッピングタスクログファイル名

各マッピングタスクログファイルのデフォルト名は、ログファイルをタイムスタンプでアーカイブするか、実行番号でアーカイブするかに基づきます。

マッピングタスクログファイルのデフォルト名は、マッピングタスク名、UID 番号、タイムスタンプ、および.log サフィックスで構成されます。実行番号でログをアーカイブする場合、デフォルトのログファイル名はマッピングタスク名、実行番号、および.log サフィックスで構成されます。

マッピングタスクの【詳細】プロパティで【マッピングタスクログ保存タイプ】を設定して、タイムスタンプまたは実行番号のどちらでアーカイブするかを指定します。実行番号でアーカイブする場合、データ統合サービスがログを上書きするまで保持する実行の数を入力できます。タイムスタンプ別にアーカイブする場合、データ統合サービスが保持するログファイル数に制限はありません。

デフォルト名を使用しない場合、別のマッピングタスクログファイル名を入力できます。

## マッピングタスクログファイルディレクトリ

データ統合サービスがマッピングタスクログを書き込むディレクトリを設定できます。マッピングタスクログファイルディレクトリをパラメータ化できます。

デフォルトでは、データ統合サービスはシステムパラメータである LogDir によって定義されたディレクトリにマッピングタスクログファイルを書き込みます。管理者は、Administrator ツールで LogDir パスを変更できます。

マッピングタスクの【詳細】プロパティで、マッピングタスクログファイルに別のディレクトリを設定できます。データ統合サービスがアクセスできないディレクトリ名を入力すると、マッピングタスクは失敗します。

グリッドでワークフローを実行する場合、マッピングタスクログディレクトリはデータ統合サービスのマスタノードを実行しているマシン上に作成されます。グリッド上のすべてのノードがアクセス可能な共有ディレクトリのマッピングタスクログファイルディレクトリを設定します。

## マッピングタスクの詳細プロパティ

マッピングタスクの【詳細】タブには、タスクでのマッピングの実行に使用されるプロパティが表示されます。

設定プロパティの値は選択が可能です。または、設定プロパティをタスク入力に割り当て、プロパティをタスク入力に割り当てる場合、ワークフローパラメータまたは変数をプロパティに割り当てることができます。ワークフローまたは変数のプロパティへの割り当ては、マッピングタスクの【入力】タブで行います。パラメータセットまたはパラメータファイルを使用して、プロパティの値を変更できます。



タスクのリカバリ戦略に値を選択する必要があります。タスクのリカバリ戦略をタスク入力に割り当てることはできません。

マッピングタスクに、以下の詳細プロパティを設定します。

### タスクリカバリストラテジ

中断された、またはエラーが発生したタスクを、データ統合サービスが再実行するかスキップするかを決定します。ワークフローでリカバリが有効になっていない場合は、データ統合サービスはタスクリカバリストラテジを無視します。

デフォルトでは、タスクは再開リカバリストラテジを使用します。

クラスタの作成タスクを使用してクラスタワークフローを作成するときは、マッピングタスクのリカバリ戦略を **【タスクの再開】** に設定します。

### クラスタ識別子

マッピングタスクを実行するクラスタを識別するクラスタ識別子文字列。

クラスタワークフローに含まれているマッピングタスクのみに **【クラスタ識別子】** プロパティを設定します。

次のいずれかのオプションを選択します。

値	説明
空欄（値なし）	マッピングと関連付けられた Hadoop 接続内に構成されたクラスタで、マッピング実行を実行します。
AutoDeploy	そのワークフローが作成したクラスタ上でマッピングを実行します。 このオプションを選択すると、クラスタの作成タスク内のクラスタ識別子プロパティは、AutoDeployCluster に設定された値を使用して入力されます。 デフォルトは AutoDeploy です。
（タスク入力に割り当てる）	クラスタの作成タスクと異なるソースからの入力を受け入れるには、このオプションを選択します。このオプションを選択した場合、マッピングタスク入力プロパティのタブのクラスタ識別子プロパティにパラメータ値を入力します。

### デフォルトの日時形式

データ統合サービスが日付データ型と文字列データ型間のデータ変換を行うために使用する日時形式。事前定義された形式から 1 つを選択するか、有効な日付形式の文字列を入力します。

プロパティは string データ型です。デフォルトは MM/DD/YYYY HH24:MI:SS です。

### 最適化レベル

データ統合サービスがマッピングに適用する最適化方式を制御します。

- 0（なし）。マッピングは最適化されません。
- 1（最小）。初期プロジェクション最適化方式がマッピングに適用されます。
- 2（ノーマル）。初期プロジェクション、初期選択、プッシュダウン、および述部の各最適化方式がマッピングに適用されます。
- 3（完全）。初期プロジェクション、プッシュダウン、述部、コストベース、準結合の各最適化方式がマッピングに適用されます。

プロパティは整数データ型です。デフォルトは 2（ノーマル）です。

### 高精度

マッピングを高精度で実行します。

「高精度」を選択すると、データ値の精度が高くなります。15 桁以上の精度など、マッピングが大きな数値を生成する場合や、正確な値を必要とする場合に「高精度」を有効にします。「高精度」を有効にする、桁数の多い数値で精度の損失を防ぐことができます。

プロパティはブールデータ型です。デフォルトは true です。

## ソート順

データ統合サービスがマッピングの文字データをソートする順序。

プロパティは文字列データ型です。デフォルトは「バイナリ」です。

## トレースレベルのオーバーライド

マッピングの各トランスフォーメーションに使用するオーバーライドトレースレベルを設定します。トレースレベルは、データ統合サービスがマッピングログファイルに送信する情報量を決定します。

以下のいずれかのトレースレベルを選択します。

- なし。トランスフォーメーションごとに設定したトレースレベルはオーバーライドされません。
- 簡易。データ統合サービスは、初期化情報、エラーメッセージ、および却下されたデータ通知をログに記録します。
- ノーマル。データ統合サービスは、初期化情報、ステータス情報、検出されたエラー、およびトランスフォーメーション行エラーでスキップされた行をログに記録します。マッピングの結果を要約しますが、個別行のレベルでは行いません。
- 詳細 - 初期化。データ統合サービスは、ノーマルトレースに加えて、初期化の詳細、インデックスファイルおよびデータファイルの名前、および詳細なトランスフォーメーション統計をログに記録します。
- 詳細 - データ。データ統合サービスにより、詳細 - 初期化トレースに加え、マッピングに渡された各行がログに記録されます。データ統合サービスにはマッピング実行のパラメータの名前と値が含まれます。また、データ統合サービスは、カラムの精度に合わせて文字列データを切り詰めた場所を示し、詳細なトランスフォーメーション統計を返します。データ統合サービスではトランスフォーメーションを処理する際にブロック中のすべての行の行データが書き込まれます。

プロパティは string データ型です。デフォルトは「通常」です。

## マッピングタスクログディレクトリ

マッピングタスクログのディレクトリです。データ統合サービスのローカルディレクトリを使用して、タスクログを保存します。パスを入力しない場合、データ統合サービスは LogDir システムパラメータで定義されたディレクトリにログファイルを書き込みます。デフォルトディレクトリは<Informatica installation directory>/tomcat/bin/disLogs/builtinhandlers です。管理者は、Administrator ツールでこのパスを変更できます。システムパラメータは、データ統合サービスのサービスプロセスプロパティにあります。

## マッピングタスクログファイル名

マッピングタスクログの名前です。ファイル名またはファイル名とディレクトリ名をこのフィールドに入力できます。データ統合サービスは、ファイル名を「マスキングタスクログファイルディレクトリ」フィールドの情報に追加します。ログファイルの保存に関して選択する方法に応じて、ログファイル名が UID およびタイムスタンプに、またはマッピング実行番号に追加されます。

## Java クラスパス

データ統合サービスがマッピングタスクを実行するときにシステムクラスパスの先頭に追加される Java クラスパス。Java トランスフォーメーションでサードパーティ製の Java パッケージ、ビルトイン Java パッケージ、またはカスタム Java パッケージを使用する場合にこのオプションを使用します。

## マッピングタスクログ保存タイプ

タイムスタンプまたはマッピングタスクの実行数でマッピングタスクログファイルを保存します。マッピングタスクログファイル名のサフィックスには、選択したオプションが反映されます。

マッピングタスクログファイルを設定する際、一度に保持するログファイルの数を制限することも、無限の数のログファイルを作成することもできます。次のいずれかの保存タイプを選択します。

- マッピングタスクタイムスタンプ。デフォルト。タイムスタンプ別のログファイルの保存。統合サービスは、タイムスタンプ別にログとラベルを無制限に保存します。各マッピングタスクログファイル名の形式は次のとおりです: <MappingTaskLogFileName>\_<UID>\_<Timestamp>.log
- マッピングタスク実行。特定数のマッピングタスクログを保存します。保存するログファイルの数を、**[次の実行でマッピングタスクログを保存]** プロパティで設定します。各マッピングタスクログファイル名の形式は次のとおりです: <MappingTaskLogFileName>\_<Run#>.log

## 次の実行でマッピングタスクログを保存

マッピングの実行数でログファイルを保存するときに保存されるログファイルの数。各ログファイル名には、実行番号が含まれます。10 個のマッピングタスクログを保存する場合、データ統合サービスは 0 から 9 までの番号が付けられたログを保存します。

## カスタムプロパティ


マッピングタスクのカスタムプロパティを定義し、プロパティの値を設定できます。

特別なケースでカスタムプロパティの適用が必要な場合があります。カスタムプロパティは、Informatica グローバルカスタマサポートから要求された場合にのみ定義します。

カスタムプロパティを定義するには、**[新規]** をクリックし、プロパティ名と初期値を入力します。カスタムプロパティ値には、ワークフローパラメータまたはワークフロー変数を入力できます。新しいワークフローパラメータまたは変数を作成することもできます。

次の図は、**[カスタムプロパティ]** パネルを示しています。

### Custom Properties:



Name	Value
Custom_Property1	NewCustomProperty (Parameter)

## 設定プロパティの変数およびパラメータの値

マッピングタスクの設定プロパティを、ワークフローの変数またはパラメータに【入力】タブで割り当てる場合は、プロパティに対する有効な変数またはパラメータの値を定義する必要があります。

次の表に、マッピングタスクの設定プロパティに対して定義できる有効な変数およびパラメータの値を示します。

プロパティ	ワークフロー変数の値	ワークフローパラメータの値
デフォルトの日時形式	有効な日付形式の文字列変数。	有効な日付形式の文字列パラメータ。
最適化レベル	<ul style="list-style-type: none"><li>- 0（なし）、1（最小）、2（ノーマル）、3（完全）のいずれかの値が設定された整数値変数。</li><li>- "0"（なし）、"1"（最小）、"2"（ノーマル）、"3"（完全）のいずれかの値が設定された文字列変数。</li><li>- 論理値変数。値が true の場合は 1（最小）に変換され、false の場合は 0（なし）に変換されます。</li></ul>	"0"（なし）、"1"（最小）、"2"（ノーマル）、"3"（完全）のいずれかの値が設定された文字列パラメータ。
高精度	<ul style="list-style-type: none"><li>- 論理値変数。</li><li>- "true"または"false"のいずれかの値が設定された文字列変数。</li><li>- 整数値変数。値が 0 の場合は false に変換され、0 以外の場合はすべて true に変換されます。</li></ul>	"true"または"false"のいずれかの値が設定された文字列パラメータ。

プロパティ	ワークフロー変数の値	ワークフローパラメータの値
ソート順	<p>次のいずれかの値が設定された文字列変数（大文字と小文字が区別されます）。</p> <ul style="list-style-type: none"> <li>- albanian</li> <li>- arabic</li> <li>- armenian</li> <li>- belarusian</li> <li>- bengali</li> <li>- binary</li> <li>- bulgarian</li> <li>- catalan</li> <li>- croatian</li> <li>- czech</li> <li>- danish</li> <li>- dutch</li> <li>- english</li> <li>- estonian</li> <li>- finnish</li> <li>- french</li> <li>- german</li> <li>- germanPhonebook</li> <li>- ギリシア語</li> <li>- ヘブライ語</li> <li>- hindi</li> <li>- hungarian</li> <li>- indonesian</li> <li>- italian</li> <li>- japanese</li> <li>- kazakh</li> <li>- korean</li> <li>- latvian</li> <li>- lithuanian</li> <li>- macedonian</li> <li>- norwegian</li> <li>- pashto</li> <li>- persian</li> <li>- polish</li> <li>- portuguese</li> <li>- romanian</li> <li>- russian</li> <li>- serbian</li> <li>- simplifiedChinese</li> <li>- slovak</li> <li>- slovenian</li> <li>- spanish</li> <li>- swedish</li> <li>- thai</li> <li>- traditionalChinese</li> <li>- traditionalSpanish</li> <li>- turkish</li> <li>- ukrainian</li> <li>- vietnamese</li> </ul>	<p>変数に対して有効な値と同じいずれかの値が設定された文字列パラメータ（大文字と小文字が区別されます）。</p>
トレースレベルのオーバーライド	<p>次のいずれかの値が設定された文字列変数（大文字と小文字が区別されます）。</p> <ul style="list-style-type: none"> <li>- terse</li> <li>- normal</li> <li>- verboseInitialization</li> <li>- verboseData</li> </ul>	<p>変数に対して有効な値と同じいずれかの値が設定された文字列パラメータ（大文字と小文字が区別されます）。</p>

プロパティ	ワークフロー変数の値	ワークフローパラメータの値
マッピングタスクログディレクトリ	データ統合サービスがアクセスできる有効なパスを含む文字列変数。デフォルトは LogDir です。	データ統合サービスがアクセスできる有効なパスを含む文字列パラメータ。デフォルトは LogDir です。
マッピングタスクログファイル名	有効なファイル名を含む文字列変数。	有効なファイル名を含む文字列パラメータ。
Java クラスパス	サードパーティまたはカスタム Java パッケージへの有効なパスを含む文字列変数。	サードパーティまたはカスタム Java パッケージへの有効なパスを含む文字列パラメータ。

#### 関連項目：

- [「ワークフロー変数のデータ型変換」 \(ページ 36\)](#)
- [「ワークフローパラメータのデータ型変換」 \(ページ 44\)](#)

## タスクで実行するマッピングの変更

マッピングタスクを設定した後、タスクで実行する別のマッピングを選択できます。

別のマッピングを選択すると、マッピングタスクに対して設定した次の情報は削除されます。

- **【入力】** タブでワークフローのパラメータまたは変数に割り当てたユーザー定義マッピングパラメータ。
- **【出力】** タブでワークフロー変数に割り当てたトランスフォーメーションの出力データ。

1. エディタでマッピングタスクを選択します。
2. **【プロパティ】** ビューで **【全般】** タブをクリックします。
3. **【参照】** をクリックし、別のマッピングを選択して **【OK】** をクリックします。
4. **【はい】** をクリックしてマッピングの変更を確定します。

選択したマッピングに対するマッピングタスクの入力と出力を設定します。

## クラスタワークフロー内のマッピングタスク

クラスタワークフローは、クラスタ作成タスクを使用してクラスタの作成を自動化し、その後にそのクラスタで 1 つ以上のマッピングを実行するワークフローです。クラスタワークフローには、1 つ以上のマッピングタスクを含める必要があります。

マッピングをマッピングタスクに追加する前に、マッピングの **【ランタイム接続】** プロパティを設定して、マッピングを実行する場所を指定します。ワークフローで作成されたクラスタでマッピングを実行するか、別のクラスタで実行するかを選択できます。

クラスタワークフローの詳細については、*Data Engineering Integration* ユーザーガイドを参照してください。

## 第 8 章

# 通知タスク

この章では、以下の項目について説明します。

- [通知タスクの概要, 111 ページ](#)
- [受信者, 112 ページ](#)
- [電子メールアドレス, 114 ページ](#)
- [電子メールのコンテンツ, 116 ページ](#)
- [通知タスクの出力, 117 ページ](#)
- [通知タスクの詳細プロパティ, 118 ページ](#)
- [通知タスクのトラブルシューティング, 118 ページ](#)

## 通知タスクの概要

通知タスクは、指定した受信者にワークフローの実行中に通知を送信するタスクです。

通知タスクでは、電子メール通知を送信することができます。例えば、ワークフローからマッピングを実行するのにかかる時間を追跡する場合は、ワークフローに通知タスクを含め、マッピングタスクが開始された日時と完了した日時を含む電子メールを送信できます。

通知タスクを設定して電子メールを送信する前に、管理者は Administrator ツールで電子メールサービスを有効にし、設定する必要があります。

ワークフローに通知タスクを追加するときは、受信者を指定し、通知のプロパティを設定します。受信者には、通知を受信する Informatica ドメインのユーザーとグループが含まれます。電子メール通知のプロパティには、電子メールアドレスや電子メールのコンテンツがあります。ワークフローのパラメータや変数を使用すると、受信者、電子メールアドレス、および電子メールのコンテンツをワークフローの実行中に動的に決めることができます。

通知タスクの実行が完了すると、出力データがワークフローの変数に渡されます。

通知タスクを設定するときは、受信者、電子メールアドレス、電子メールのコンテンツ、およびタスクからワークフローに渡される出力データを指定します。さらに、実行時にタスクで使用する詳細プロパティも設定します。

# 受信者

受信者には、ワークフローの実行中に通知を受信する Informatica ドメインのユーザーとグループが含まれます。受信者は**【通知】** タブで選択します。

ネイティブセキュリティドメインと LDAP セキュリティドメインのユーザーおよびグループを選択できます。選択したユーザーおよびグループは、次の形式で Developer tool に表示されます。

<security domain name>\<user login name or group name>

電子メール通知の受信者としてユーザーを設定した場合、通知タスクにより、ユーザーアカウントのプロパティで指定する電子メールアドレスのユーザーに電子メールが送信されます。電子メール通知の受信者としてグループを設定した場合は、通知タスクにより、ユーザーアカウントのプロパティで指定する電子メールアドレスのグループの全ユーザーに電子メールが送信されます。

ユーザーアカウントの電子メールアドレスが無効な場合、通知タスクが実行され、配信不能なアドレスであることがドメインに通知されます。通知タスクにより、電子メールサービスが指定するアドレスに電子メールが送信され、配信失敗を示します。

ネイティブユーザーアカウントの電子メールアドレスは、Administrator ツールを使用して入力できます。LDAP ユーザーアカウントの電子メールアドレスは、Administrator ツールで入力することはできません。LDAP ディレクトリサービスからインポートされたユーザーアカウント情報に電子メールアドレスが含まれている場合、通知タスクは、その電子メールアドレスのユーザーに電子メールを送信できます。

受信者としてユーザーを選択した後に、そのユーザーが属するグループを選択すると、ユーザーが重複して選択されることになります。通知タスクが電子メールを送信すると、重複する受信者は、電子メールサーバーでの重複する受信者の処理方法に応じて、複数の電子メールを受信するか、または 1 つだけ受信します。

通知タスクの電子メールプロパティを設定するときは、アドレスフィールドにユーザーやグループの名前を入力できるほか、動的な受信者を設定することもできます。動的な受信者とは、ワークフローのパラメータまたは変数で定義したユーザーおよびグループです。

## 受信者の選択

ユーザーやグループに通知を送信する場合は、そのユーザーおよびグループを受信者リストに追加します。

受信者リストへのユーザーやグループの追加は、通知タスクの受信者リストまたは電子メールのプロパティで行います。どちらかでユーザーやグループを追加すると、もう一方にも表示されます。ユーザーやグループの名前を入力できるほか、電子メールのプロパティでは動的な受信者を設定することもできます。

ユーザーやグループを追加するときは、検索フィルタを使用できます。ユーザーまたはグループを検索するための文字列を入力します。この文字列にはワイルドカード文字を使用できます。検索文字列を含むすべての名前が返されます。文字列の大文字と小文字は区別されません。

1. エディタで通知タスクを選択します。
2. **【プロパティ】** ビューで **【通知】** タブをクリックします。
3. **【受信者】** を選択します。
4. **【選択】** をクリックします。  
**【ユーザーおよびグループの選択】** ダイアログボックスが表示されます。
5. ユーザーまたはグループを検索するためのフィルタ条件を入力します。  
グループ内のユーザーを表示するには、グループを選択し、**【ユーザーをグループで表示】** をクリックします。
6. ユーザーまたはグループの名前を選択します。  
複数の名前を選択する場合は、Ctrl キーまたは Shift キーを使用します。
7. **【OK】** をクリックします。



## 受信者の通知タイプの設定

電子メール通知を受信する受信者を設定するには、受信者の電子メールアドレスのフィールドを選択します。

1. エディタで通知タスクを選択します。
2. **【プロパティ】** ビューで **【通知】** タブをクリックします。
3. **【受信者】** を選択します。
4. 受信者リストで、ユーザー、グループ、または動的な受信者を選択します。
5. **【電子メール】** カラムで、**【宛先】**、**【Cc】**、または **【Bcc】** を選択します。

## 受信者名の入力

通知タスクの電子メールのプロパティを設定するときは、アドレスフィールドにユーザーやグループの名前を入力できます。

アドレスフィールドにユーザーやグループの名前を入力する際の形式は次のとおりです。

<security domain name>\<user login name or group name>

例えば、ネイティブセキュリティドメインの adietrich というログイン名のユーザーを指定するには、Native \adietrich と入力します。

セキュリティドメイン名を指定しなかった場合は、ネイティブセキュリティドメインが使用されます。

1. エディタで通知タスクを選択します。
2. **【プロパティ】** ビューで **【通知】** タブをクリックします。
3. **【通知タイプ】** で **【電子メール】** を選択します。
4. いずれかのアドレスフィールドに、所定の形式でユーザーまたはグループの名前を入力します。  
受信者名を複数入力する場合は、セミコロンで区切ります。

## 動的な電子メール受信者

ワークフローのパラメータや変数を使用すると、電子メールの受信者を動的に定義できます。

電子メールのプロパティで受信者を設定するときに、**【動的な受信者】** タブを使用して、ユーザーやグループの受信者を定義するワークフローのパラメータや変数を選択できます。パラメータや変数の名前をアドレスフィールドに入力することはできません。

次の表に、動的な受信者を定義する場合に使用できるパラメータと変数のタイプを示します。

パラメータまたは変数のタイプ	説明
ワークフローパラメータ	文字列のワークフローパラメータを使用して、ワークフローの実行時のユーザーまたはグループの名前をパラメータファイルで定義することができます。
システムワークフロー変数	UserName システムワークフロー変数を使用して、ワークフローを実行するユーザーに電子メールを送信することができます。
ユーザー定義ワークフロー変数	ユーザー定義の文字列のワークフロー変数を使用して、ワークフローにおけるユーザーまたはグループの名前を割り当てタスクで動的に割り当てることができます。

パラメータファイルまたは割り当てタスクでユーザーまたはグループの名前を定義するときは、値として単一のユーザーまたはグループの名前を入力します。値の構文は次のとおりです。

<security domain name>\<user login name or group name>

例えば、ネイティブセキュリティドメインの Developers グループを指定するには、Native\Developers と入力します。

セキュリティドメイン名を指定しなかった場合は、ネイティブセキュリティドメインが使用されます。Data Integration Service では、指定された名前をグループのリストで先に検索し、その名前のグループが見つからなかった場合にユーザーのリストで検索します。

## 動的な電子メール受信者の設定

ワークフローのパラメータや変数を使用すると、動的な電子メール受信者を設定できます。

1. エディタで通知タスクを選択します。
2. **【プロパティ】** ビューで **【通知】** タブをクリックします。
3. **【通知タイプ】** で **【電子メール】** を選択します。
4. **【宛先】** をクリックします。  
**【メールプロパティ】** ダイアログボックスが表示されます。
5. **【受信者】** を選択し、**【動的な受信者】** をクリックします。
6. **【新規】** をクリックします。  
受信者リストに空の行が追加されます。
7. **【受信者】** カラムをクリックし、既存のパラメータまたは変数を選択するか、新しいパラメータまたは変数を作成します。
  - ワークフローのパラメータまたは変数の名前を選択します。
  - **【新しいパラメータ】** または **【新しい変数】** をクリックします。 **【パラメータの追加】** ダイアログボックスまたは **【変数の追加】** ダイアログボックスで、パラメータまたは変数の名前とタイプを入力します。パラメータの場合はデフォルトのユーザー名またはグループ名を入力し、変数の場合は初期のユーザー名またはグループ名を入力します。ワークフローのパラメータまたは変数が作成され、動的な受信者のリストに追加されます。
8. **【電子メール】** カラムで、**【宛先】**、**【Cc】**、または **【Bcc】** を選択します。
9. **【OK】** をクリックします。

ワークフローのパラメータまたは変数が、通知タスクの受信者リストの **【動的な受信者】** の下に表示されます。

## 電子メールアドレス

電子メールの受信者としてユーザーやグループを指定することに加え、通知タスクの電子メールを受信する電子メールアドレスを入力することもできます。電子メールアドレスは **【通知】** タブで入力します。

有効なものであれば、任意の電子メールアドレスを入力できます。電子メールアドレスを複数入力する場合は、セミコロンで区切ります。ワークフローのパラメータや変数を使用すると、電子メールアドレスを動的に定義できます。

無効な電子メールアドレスを入力すると、通知タスクが実行され、ドメインに対し電子メールアドレスに配信できないことが通知されます。通知タスクにより、電子メールサービスが指定するアドレスに電子メールが送信され、配信失敗を示します。

## 電子メールアドレスの入力

通知タスクは、電子メールのプロパティに入力されたあらゆる有効な電子メールアドレスに電子メールを送信できます。

1. エディタで通知タスクを選択します。
2. **【プロパティ】** ビューで **【通知】** タブをクリックします。
3. **【通知タイプ】** で **【電子メール】** を選択します。
4. **【プロパティ】** 領域の適切なアドレスフィールドに、完全修飾の電子メールアドレスを入力します。  
電子メールアドレスを複数入力する場合は、セミコロンで区切ります。

## 動的な電子メールアドレス

ワークフローのパラメータや変数を使用すると、電子メールアドレスを動的に定義できます。

電子メールのプロパティで電子メールアドレスを入力するときに、**【動的な電子メールアドレス】** タブを使用して、電子メールアドレスを定義するパラメータや変数を選択できます。パラメータや変数の名前をアドレスフィールドに入力することはできません。

次の表に、動的な電子メールアドレスを定義する場合に使用できるパラメータと変数のタイプを示します。

パラメータまたは変数のタイプ	説明
ワークフローパラメータ	文字列のワークフローパラメータを使用して、ワークフローの実行時の電子メールアドレスをパラメータファイルで定義することができます。
システムワークフロー変数	UserName システムワークフロー変数を使用して、ワークフローを実行するユーザーに電子メールを送信することができます。
ユーザー定義ワークフロー変数	ユーザー定義の文字列のワークフロー変数を使用して、ワークフローにおける電子メールアドレスを割り当てタスクで割り当てることができます。

パラメータファイルまたは割り当てタスクで電子メールアドレスを定義するときは、値として単一の電子メールアドレスを入力します。

## 動的な電子メールアドレスの設定

ワークフローのパラメータや変数を使用すると、動的な電子メールアドレスを設定できます。

1. エディタで通知タスクを選択します。
2. **【プロパティ】** ビューで **【通知】** タブをクリックします。
3. **【通知タイプ】** で **【電子メール】** を選択します。
4. **【宛先】** をクリックします。  
**【メールプロパティ】** ダイアログボックスが表示されます。
5. **【電子メールアドレス】** を選択し、**【動的な電子メールアドレス】** をクリックします。
6. **【新規】** をクリックします。  
受信者リストに空の行が追加されます。
7. **【受信者】** カラムをクリックし、既存のパラメータまたは変数を選択するか、新しいパラメータまたは変数を作成します。

- ワークフローのパラメータまたは変数の名前を選択します。
  - **【新しいパラメータ】** または **【新しい変数】** をクリックします。 **【パラメータの追加】** ダイアログボックスまたは **【変数の追加】** ダイアログボックスで、パラメータまたは変数の名前とタイプを入力します。パラメータの場合はデフォルトの電子メールアドレスを入力し、変数の場合は初期の電子メールアドレスを入力します。
8. **【電子メール】** カラムで、**【宛先】**、**【Cc】**、または **【Bcc】** を選択します。
  9. **【OK】** をクリックします。

## 電子メールのコンテンツ

電子メールのコンテンツには、電子メールの件名と本文が含まれます。電子メールのコンテンツは **【通知】** タブで入力します。ワークフローのパラメータや変数を使用すると、電子メールのコンテンツを動的に定義できます。

### 動的な電子メールのコンテンツ

電子メールの件名や本文でワークフローのパラメータや変数を使用すると、コンテンツを動的に定義できます。

電子メールのプロパティで **【電子メールのコンテンツ】** を選択した場合は、ワークフローのパラメータや変数の名前を選択できます。

**【通知】** タブで通知タイプとして **【電子メール】** を選択した場合は、件名または本文のフィールドにワークフローのパラメータや変数の名前を入力できます。件名または本文にパラメータや変数の名前を入力する際は、所定の構文に従う必要があります。

例えば、MappingErrorRows という名前のワークフロー変数を作成し、その変数にマッピングタスクのエラー行の数を示す出力値を割り当てるとします。通知タスクの本文に次のテキストを入力します。

Mapping failed to write \${var:MappingErrorRows} rows to the target.

次の表に、動的なコンテンツを定義する場合に使用できるパラメータと変数のタイプを示します。

パラメータまたは変数のタイプ	説明
ワークフローパラメータ	任意の型のワークフローパラメータを使用して、ワークフローの実行時の電子メールのコンテンツをパラメータファイルで定義することができます。
システムワークフロー変数	任意のシステムワークフロー変数を使用して、システムワークフロー変数の値を電子メールのコンテンツに含めることができます。
ユーザー定義ワークフロー変数	任意のデータ型のワークフロー変数を使用して、他のタスクで生成された出力値を電子メールのコンテンツに含めることができます。また、割り当てタスクで変数に割り当てられた値を電子メールのコンテンツに含めることもできます。

#### 関連項目：

- [「式および文字列のパラメータ名」 \(ページ 41\)](#)
- [「式および文字列の変数名」 \(ページ 34\)](#)

## 電子メールのコンテンツの入力

電子メールのコンテンツを設定するときは、電子メールの件名と本文のテキストを入力します。ワークフローのパラメータや変数を使用すると、動的な電子メールのコンテンツを設定できます。

1. エディタで通知タスクを選択します。
2. **【プロパティ】** ビューで **【通知】** タブをクリックします。
3. **【通知タイプ】** で **【電子メール】** を選択します。
4. **【件名】** をクリックします。  
**【メールプロパティ】** ダイアログボックスが表示されます。
5. **電子メールのコンテンツ** ビューで、件名および本文のフィールドにテキストを入力します。
6. 既存のワークフローのパラメータまたは変数を使用して動的なコンテンツを設定する場合は、件名または本文のフィールドを選択し、既存のワークフローのパラメータまたは変数をダブルクリックします。  
所定の構文に従って、パラメータまたは変数がフィールドに追加されます。
7. 新しいワークフローのパラメータまたは変数を使用して動的なコンテンツを設定する場合は、**【新しいパラメータ】** または **【新しい変数】** をクリックします。  
**【パラメータの追加】** ダイアログボックスまたは **【変数の追加】** ダイアログボックスで、パラメータまたは変数の名前とタイプを入力します。パラメータの場合はデフォルト値を入力し、変数の場合は初期値を入力します。ワークフローのパラメータまたは変数が作成され、電子メールのコンテンツのフィールドに追加されます。
8. **【OK】** をクリックします。

## 通知タスクの出力

通知タスクの出力は、通知タスクからワークフロー変数に渡されるデータです。通知タスクの出力には、全般出力が含まれます。

通知タスクを設定するときに、**【出力】** タブで、ワークフロー変数に割り当てるタスクの出力値を指定します。Data Integration Service は、通知タスクが完了または失敗したときに通知タスクの出力値をワークフロー変数にコピーします。タスクが強制終了された場合には、タスク出力値はワークフロー変数にコピーされません。

例えば、ある通知タスクが、タスクの実行が開始された時刻を示す出力値を生成するとします。この出力データにワークフローから直接アクセスすることはできません。このデータをワークフローの残りのオブジェクトで使用するには、NotificationStartTime という名前のワークフロー変数に開始時刻の出力を割り当て、その NotificationStartTime ワークフロー変数を条件付きシーケンスフローの式で使用します。通知タスクが指定の時刻よりも前に実行された場合は、Data Integration Service はワークフローの次のオブジェクトを実行します。

タスクの開始時刻、終了時刻、およびタスクの実行が成功したかどうかなど、全般出力にはすべてのタスクで生成される出力データが含まれます。

### 関連項目：

- [「タスク出力」 \(ページ 28\)](#)
- [「タスク出力を使用した値の割り当て」 \(ページ 31\)](#)

## 通知タスクの詳細プロパティ

通知タスクの【詳細】タブには、タスクのリカバリストラテジが含まれます。

以下の通知タスクの詳細プロパティを設定します。

### タスクリカバリストラテジ

中断された、またはエラーが発生したタスクを、データ統合サービスが再実行するかスキップするかを決定します。ワークフローでリカバリが有効になっていない場合は、データ統合サービスはタスクリカバリストラテジを無視します。

デフォルトでは、タスクは再開リカバリストラテジを使用します。

## 通知タスクのトラブルシューティング

次に示す状況に対する解決方法は、通知タスクのトラブルシューティングで役に立つことがあります。

**通知タスクが失敗し、「応答の読み取り中に例外発生」というメッセージが表示されます。**

このメッセージは SMTP メールサーバーから送られたもので、電子メールサーバーのプロパティが Administrator ツールで正しく設定されていなかったことを示している可能性があります。例えば、電子メールサーバープロパティで、SMTP サーバーが SSL セキュリティを使用するように指定されている可能性がありますしかし、指定された SMTP サーバーのポート番号は、SSL ポート番号ではなく、TLS ポート番号になります。

## 第 9 章

# ゲートウェイ

この章では、以下の項目について説明します。

- [ゲートウェイの概要, 119 ページ](#)
- [排他ゲートウェイ, 120 ページ](#)
- [包含ゲートウェイ, 121 ページ](#)
- [デフォルトシーケンスフロー, 123 ページ](#)
- [シーケンスフローの複数のブランチへの分割, 123 ページ](#)
- [ブランチの 1 つのシーケンスフローへのマージ, 124 ページ](#)

## ゲートウェイの概要

ゲートウェイは 1 つのシーケンスフローを複数のシーケンスフローに分割したり、複数のシーケンスフローを 1 つのシーケンスフローにマージしたりします。データ統合サービスは、実行時にシーケンスフローを評価し、指定した条件を満たすシーケンスフローのオブジェクトを実行します。

ゲートウェイはペアでワークフローに追加します。最初のゲートウェイは、前段階のワークフローオブジェクトからのシーケンスフローを分割し、選択した複数のオブジェクトに接続します。2 番目のゲートウェイはシーケンスフローをマージして、1 つのシーケンスフローをワークフローの次のオブジェクトに接続します。最初のゲートウェイは出力ゲートウェイです。2 番目のゲートウェイは入力ゲートウェイです。1 つのゲートウェイは、少なくとも 2 つのシーケンスフローに接続する必要があります。

各シーケンスフローは、ワークフローデータがたどることができるブランチを表します。ブランチでは、複数のオブジェクトを連続して接続できます。各ブランチの最終シーケンスフローは入力ゲートウェイに接続します。

出力ゲートウェイに作成したブランチは、同じタイプの入力ゲートウェイにマージされる必要があります。各ブランチが入力ゲートウェイにマージされる前に、ブランチの複数のワークフローオブジェクトを接続できます。

排他ゲートウェイと包含ゲートウェイを 1 つのワークフローに追加できます。ゲートウェイの間で 1 つのブランチのオブジェクトを実行するには、排他ゲートウェイを追加します。複数のブランチで並行してオブジェクトを実行するには、包含ゲートウェイを追加します。例えば、マッピングが正常に実行された場合はあるブランチへワークフローが進み、マッピングが失敗した場合は別のブランチに進むことを決定する場合は、排他ゲートウェイを使用します。1 つのアプリケーションから並行して一連のマッピングを実行する場合は、包含ゲートウェイを使用します。

2 つのゲートウェイの間のブランチに排他ゲートウェイと包含ゲートウェイを追加できます。2 つのゲートウェイの間のブランチにゲートウェイを追加する場合、ゲートウェイをペアで追加します。例えば、出力包含ゲートウェイと入力包含ゲートウェイを 2 つのゲートウェイの間のブランチに追加します。または、出力排他ゲ



ートウェイと入力排他ゲートウェイをブランチに追加します。ブランチに追加するゲートウェイは、ネストされたゲートウェイと呼ばれます。

出力ゲートウェイに作成したブランチは、同じタイプの入力ゲートウェイにマージされる必要があります。例えば、出力排他ゲートウェイからのブランチは、入力排他ゲートウェイにマージされる必要があります。各ブランチが入力ゲートウェイにマージされる前に、ブランチの複数のワークフローオブジェクトを接続できます。

出力ゲートウェイで作成したシーケンスフローに条件を定義します。データ統合サービスにより出力排他ゲートウェイで条件が評価される場合、true と評価された条件を含む最初のシーケンスフローが実行されます。データ統合サービスにより出力包含ゲートウェイで条件が評価される場合、true と評価された条件を含むすべてのシーケンスフローが実行されます。包含ゲートウェイのシーケンスフローは同時に実行されます。

出力ゲートウェイごとにデフォルトシーケンスフローを選択する必要があります。デフォルトシーケンスフローを選択することで、シーケンスフローの条件がすべて false に評価された場合でも、データが出力ゲートウェイから入力ゲートウェイに流れます。データ統合サービスでは、他のシーケンスフローに対する条件が false に評価された場合、排他ゲートウェイのデフォルトシーケンスフローが実行されます。包含ゲートウェイでは、デフォルトシーケンスフローは常に実行されます。デフォルトシーケンスフローに条件を定義する必要はありません。

## 排他ゲートウェイ

排他ゲートウェイを使用すると、1つのシーケンスフローから複数のブランチを作成して、単一のブランチでオブジェクトを実行できます。データ統合サービスは、true と評価されたシーケンスフロー条件を含む最初のブランチでオブジェクトを実行します。

データ統合サービスは、出力ゲートウェイプロパティで指定した順序で各シーケンスフローに対する条件を評価します。条件が true と評価されると、そのシーケンスフローが示すブランチが実行され、他のブランチは評価されません。条件が false と評価されると、そのブランチはスキップされ、次のシーケンスフローに対する条件が評価されます。ブランチのオブジェクトが完了すると、データ統合サービスはデータを入力排他ゲートウェイに渡します。

ゲートウェイプロパティでデフォルトシーケンスフローを指定します。ゲートウェイプロパティで設定した順序に関係なく、データ統合サービスはデフォルトシーケンスフローを最後に評価します。出力排他ゲートウェイで実行するブランチがなくなると、デフォルトシーケンスフローが示すブランチが実行されます。

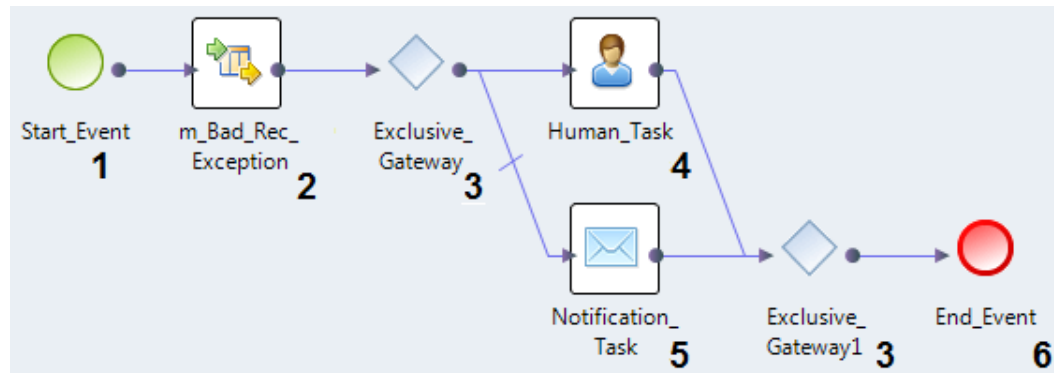
## 排他ゲートウェイの例

互いに排他的に実行されるブランチを作成するには、排他ゲートウェイをワークフローに追加します。排他ゲートウェイが含まれるワークフローを実行すると、データ統合サービスはゲートウェイ間の1つのブランチのタスクを実行します。実行時にワークフローが進むブランチを決定するには、出力シーケンスフローに対する条件を設定します。

前段階のワークフローオブジェクトからの結果に応じて、ワークフローが進むパスが決まる場合に、排他ゲートウェイを使用できます。例えば、例外レコードを識別するマッピングタスクと例外レコードをデータスチュワードに配布するヒューマンタスクを含むワークフローを設定します。マッピングタスクのマッピングで例外が識別されないと、ヒューマンタスクはレコードを配布できません。排他ゲートウェイを使用してワークフローに2つのブランチを作成します。一方のブランチには、例外レコードを配布できるヒューマンタスクが含まれます。他方のブランチには、マッピングソースデータに例外レコードがない場合にデータスチュワードに電子メールを送信する通知タスクが含まれます。



次の図に、排他ゲートウェイの間にブランチが含まれるワークフローを示します。



ワークフローには次のオブジェクトが含まれます。

1. 開始イベント。開始イベントはワークフローを開始します。
2. マッピングタスク。マッピングタスクはデータセット内で例外レコードを検出するマッピングを識別します。マッピングタスク出力には、*exceptionLoadCount* 変数値が含まれます。ヒューマンタスクは、*exceptionLoadCount* 値を使用してデータスチュワードに配布する例外レコード数を決定します。
3. 排他ゲートウェイ。最初のゲートウェイはシーケンスフローを2つのブランチに分割します。一方のブランチにはヒューマンタスクが含まれます。他方のブランチには通知タスクが含まれます。2番目のゲートウェイはブランチを1つのシーケンスフローにマージします。  
**注:** ヒューマンタスクに接続するシーケンスフローに1つの条件を定義します。マッピングタスクが生成する *exceptionLoadCount* 変数値がゼロより大きい場合、条件が *true* と評価されるように設定します。
4. ヒューマンタスク。ヒューマンタスクは前のマッピングで識別された例外レコードを読み取り、分析のためにデータスチュワードに配布します。
5. 通知タスク。通知タスクはデータスチュワードに電子メールを送信し、ヒューマンタスクが実行されなかったことを通知します。  
通知タスクに接続するシーケンスフローはデフォルトシーケンスフローです。データ統合サービスは、ヒューマンタスクシーケンスフローに対する条件が *false* と評価された場合に通知タスクを実行します。
6. 終了イベント。終了イベントはワークフローを終了します。

## 包含ゲートウェイ

シーケンスフローから複数のブランチを作成し、1つ以上のブランチで並行してオブジェクトを実行するには、包含ゲートウェイを使用します。データ統合サービスは *true* と評価されたシーケンスフロー条件を含むすべてのブランチでオブジェクトを実行します。

データ統合サービスは、ブランチのオブジェクトを実行する前に、各シーケンスフローに対する条件を評価します。条件が *true* と評価されると、データ統合サービスはそのシーケンスフローが示すブランチに進みます。各ブランチのオブジェクトは同時に実行されます。すべてのブランチのオブジェクトが完了すると、データ統合サービスはデータを入力包含ゲートウェイからワークフローの次のオブジェクトに渡します。

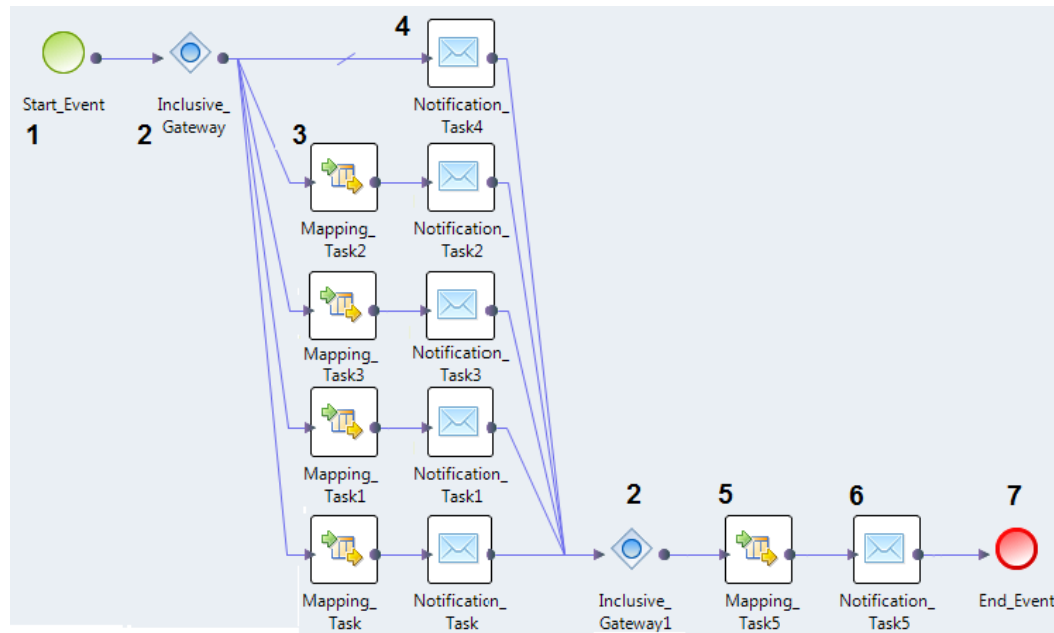
ゲートウェイプロパティでデフォルトシーケンスフローを指定します。デフォルトシーケンスフローに条件を定義する必要はありません。包含ゲートウェイでは、デフォルトシーケンスフローは常に実行されます。

## 包含ゲートウェイの例

複数のタスクを並行して実行するには、ワークフローに包含ゲートウェイを追加します。例えば、複数ブランチを含む包含ゲートウェイを設定し、各ブランチにマッピングタスクを追加できます。ワークフローが実行されると、データ統合サービスはタスクで識別されたマッピングを同時に実行します。

膨大な量のデータを処理する必要がある場合に、複数のマッピングを並行して設定することができます。タスクでは、それぞれ別の場所のデータを読み取るマッピングを識別し、そのデータをウェアハウスに書き込むことがあります。包含ゲートウェイで定義されたブランチにマッピングタスクを追加します。データ統合サービスは各マッピングを個別のジョブとして扱います。データ統合サービスをグリッドで実行されるように設定した場合、マッピングジョブはグリッド上の異なるノードに割り当てられます。Hadoop 環境でマッピングを実行するように設定した場合、データ統合サービスは Hadoop クラスタにマッピングをプッシュします。

次の図に、包含ゲートウェイの間に複数のマッピングタスクを含むワークフローを示します。



ワークフローには次のオブジェクトが含まれます。

1. 開始イベント。開始イベントはワークフローを開始します。
2. 包含ゲートウェイ。2つのゲートウェイはワークフローを複数のブランチに分割し、それらのブランチを1つのフローにマージします。
3. マッピングタスク。ワークフローのブランチには、一連のマッピングタスクが含まれます。各タスクにより、別のデータソースを読み取るマッピングが識別されます。マッピングは共通のデータターゲットに書き込みます。
4. 通知タスク。各ブランチには、対応するマッピングが実行されるとデータ所有者に電子メールを送信する通知タスクが含まれます。  
デフォルトシーケンスフローはマッピングタスクに接続しません。このシーケンスフローは、データ統合サービスが出力ゲートウェイに到達したことをワークフロー所有者に通知する通知タスクに接続します。
5. マッピングタスク。最後のマッピングタスクには、前段階のマッピングがターゲットデータベーステーブルに書き込んだレコード数を確認する式トランスフォーメーションが含まれます。
6. 通知タスク。最後の通知タスクは、ワークフロータスクが完了したことを通知する電子メールをワークフロー開発者に送信します。
7. 終了イベント。終了イベントはワークフローを終了します。

**注:** マッピングタスクに Hadoop 環境で実行するように設定されたマッピングが含まれている場合は、マッピングタスク出力をワークフロー変数に割り当てないでください。Hadoop 環境で実行されるマッピングからはソース、ターゲット、またはエラー行の合計数を取得できません。マッピングタスクに Hadoop 環境で実行されるマッピングが含まれている場合、そのタスク出力にはゼロ (0) の値が含まれています。

## デフォルトシーケンスフロー

出力ゲートウェイを他のオブジェクトにリンクするシーケンスフローを作成する場合、デフォルトシーケンスフローを指定します。デフォルトシーケンスフローを指定すると、データ統合サービスは出力シーケンスフローで実行されるオブジェクトを常に識別できます。

データ統合サービスはゲートウェイのタイプごとに異なる方法でデフォルトシーケンスフローを使用します。排他ゲートウェイでは、他のすべてのシーケンスフローに対する条件が false と評価される場合、データ統合サービスはデフォルトシーケンスフローでオブジェクトを実行します。包含ゲートウェイでは、他のすべてのシーケンスフローに対する条件に関係なく、データ統合サービスはデフォルトシーケンスフローでオブジェクトを実行します。

デフォルトシーケンスフローに対する条件を定義する必要はありません。実行時、データ統合サービスはデフォルトシーケンスフローに対する条件を無視します。シーケンスフローに条件が含まれ、そのシーケンスフローがデフォルトとして選択された場合、Developer tool にはその条件が読み取り専用として表示されます。

**注:** ゲートウェイが表すビジネス決定によっては、ワークフローデータに対し作業を先に進める必要がない場合があります。ゲートウェイ間のオブジェクトが実行できない場合にワークフローを続行するには、出力ゲートウェイを入力ゲートウェイに接続するシーケンスフローを作成します。そのシーケンスフローをデフォルトシーケンスフローに選択します。

## シーケンスフローの複数のブランチへの分割

1つのシーケンスフローを複数のシーケンスフローに分割するには、ゲートウェイを使用します。ゲートウェイを複数のオブジェクトに接続します。各接続が1つのシーケンスフローであり、そこからワークフローの1つのブランチが始まります。

1. ゲートウェイをワークフローに追加します。
2. ワークフローにオブジェクトを追加して、データ統合サービスが各ブランチで実行するアクションを表します。
3. オブジェクトを設定します。
4. ゲートウェイを各オブジェクトに接続します。エディタか、または **【シーケンスフロー】** タブを使用します。
  - エディタでは、ゲートウェイを選択し、各オブジェクトにドラッグします。
  - **【シーケンスフロー】** タブでは、**【新規】** をクリックします。**【ワークフローオブジェクトの接続】** ダイアログボックスで、オブジェクトを選択し、ゲートウェイに接続します。
5. ゲートウェイを選択します。
6. **【プロパティ】** ビューで **【シーケンスフロー】** タブをクリックします。
7. デフォルトのシーケンスフローを設定するには、シーケンスフローを選択し、**【デフォルトとして設定】** をクリックします。

8. デフォルトシーケンスフローではない各出力シーケンスフローで条件式を定義します。
  - シーケンスフローの条件エディタを表示するには、**【条件】** カラムの矢印をクリックします。
  - 式にはワークフローパラメータと変数を追加できます。条件エディタの**【入力】** タブでワークフローパラメータまたは変数を選択するか、所定の構文で条件の中にパラメータ名または変数名を入力します。
9. 矢印を使用してデータ統合サービスがシーケンスフローを評価する順序を定義します。

ブランチを 1 つのシーケンスフローにマージするには、別のゲートウェイを使用します。同じタイプのゲートウェイを使用して、ブランチを作成し、ブランチをマージします。

#### 関連項目：

- [「条件付きシーケンスフロー」 \(ページ 15\)](#)

## ブランチの 1 つのシーケンスフローへのマージ

ワークフローのブランチを 1 つのシーケンスフローにマージするには、ゲートウェイを使用します。各ブランチの最終オブジェクトを、追加したゲートウェイに接続します。

1. ゲートウェイをワークフローに追加します。

ブランチを作成したゲートウェイのタイプを選択します。
  2. 各ブランチの最終オブジェクトをゲートウェイに接続します。
- 最終オブジェクトを接続後、ワークフローで次に実行する 1 つのオブジェクトにゲートウェイを接続できます。

## 第 10 章

# ワークフローリカバリ

この章では、以下の項目について説明します。

- [ワークフローリカバリの概要, 125 ページ](#)
- [リカバリ可能なワークフローの状態, 126 ページ](#)
- [リカバリ不能なワークフローの状態, 126 ページ](#)
- [タスクのエラーおよびリカバリ, 126 ページ](#)
- [ワークフローリカバリストラテジ, 127 ページ](#)
- [タスクリカバリストラテジ, 129 ページ](#)
- [ワークフローおよびタスクリカバリのルールとガイドライン, 132 ページ](#)
- [リカバリの設定手順, 133 ページ](#)
- [ワークフローリカバリ中のパラメータと変数の値, 134 ページ](#)
- [ワークフローリカバリログ, 135 ページ](#)
- [再開のマッピングの開発手順, 135 ページ](#)
- [ワークフローインスタンスのリカバリ, 137 ページ](#)
- [中断後のワークフローの状態のサマリ, 137 ページ](#)
- [ワークフローリカバリの例, 138 ページ](#)

## ワークフローリカバリの概要

ワークフローリカバリとは、中断ポイントからワークフローインスタンスを完了することです。エラーが発生した場合、ユーザがワークフローインスタンスをキャンセルまたは強制終了した場合、またはデータ統合サービスプロセスが予期せずにシャットダウンした場合は、実行中のワークフローインスタンスが中断されることがあります。

データ統合サービスは、予期しないシャットダウン後にサービスを再開する場合、前のワークフローの状態にリカバリしようとしています。デフォルトでは、データ統合サービスはコマンドタスク、マッピングタスク、または通知タスクの実行中に停止したワークフローインスタンスをリカバリしません。また、データ統合サービスは、ワークフローインスタンスまたはワークフローインスタンスで実行中のタスクをキャンセルした場合、デフォルトでは、ワークフローインスタンスをリカバリできません。ワークフローのリカバリオプションを設定して、データ統合サービスによる上記の場合のワークフローインスタンスのリカバリを有効にできます。

ワークフローオプションを設定する際に、ワークフローを手動リカバリまたは自動リカバリに設定できます。自動リカバリに設定すると、データ統合サービスは、人の操作なしに、中断ポイントからワークフローを再開します。手動リカバリに設定すると、手動でワークフローを再開できます。

データ統合サービスでは、ワークフロー停止時に実行中だったタスクに対しワークフローが指定するタスクリカバリストラテジも考慮されます。タスクリカバリストラテジにより、ワークフロー実行中に中断されたタスクをデータ統合サービスで完了するかどうかが決まります。

**注:** 一部のエラーはリカバリ不能です。ワークフローエラーからのワークフローインスタンスのリカバリはできません。強制終了状態のワークフローはリカバリできません。ワークフローインスタンスがリカバリ可能で、ワークフローメタデータが変更され、ワークフローアプリケーションが再デプロイされる場合、そのワークフローインスタンスはリカバリ不能になります。

## リカバリ可能なワークフローの状態

キャンセルしたワークフローインスタンスおよびリカバリ可能なエラーで中断されたワークフローインスタンスをリカバリできます。ワークフローをリカバリするには、まずワークフローのリカバリを有効にします。

Administrator ツールの **【モニタ】** タブから、またはコマンドラインからワークフローインスタンスをキャンセルできます。リカバリ可能なエラーには、ワークフローを実行するデータ統合サービスプロセスの中断が含まれます。

## リカバリ不能なワークフローの状態

ワークフローエラーにより、実行中のワークフローインスタンスが中断されることがあります。ワークフローエラーはリカバリ不能です。

ワークフローエラーは、データ統合サービスが次のアクションを実行できない場合に発生します。

- パラメータを解決するために、ワークフロー実行の開始時にパラメータファイルを読み込む。例えば、ワークフローのトレースレベルプロパティに割り当てられているパラメータに無効な値が含まれる場合、ワークフローエラーが発生します。
- ワークフローのパラメータおよび変数の値をタスク入力にコピーする。例えば、文字列パラメータ値が整数タスクの入力に割り当てられていて、文字列値に数値が含まれない場合、ワークフローエラーが発生します。

ワークフローのエラーでは、エラーを修正するために、ワークフロー定義の変更またはパラメータファイルのパラメータ値の変更が必要になります。ワークフローエラーが発生すると、ワークフローインスタンスは直ちに失敗します。ワークフローでリカバリが有効になっている場合でも、ワークフローインスタンスはリカバリできません。

**注:** アクティブなシーケンスフローが終結イベントに到達して終了したワークフローは、強制終了状態に入ります。このワークフローインスタンスをリカバリすることはできません。

## タスクのエラーおよびリカバリ

タスクエラーが発生すると、リカバリ戦略次第で、実行中のワークフローインスタンスが中断される場合と、データ統合サービスによりワークフローインスタンス実行中にタスクがスキップされる場合があります。

タスクのエラーはリカバリ可能またはリカバリ不能です。リカバリ可能なタスクエラーとは、ワークフロー定義を変更せずに修正できるエラーです。リカバリ不能なタスクエラーの場合、ワークフロー定義を変更するか、

またはパラメータファイルでパラメータ値を変更する必要があります。ワークフローのリカバリを有効にした場合、リカバリ可能なタスクエラーにより中断されたワークフローインスタンスをリカバリできます。

ワークフロータスクでは、リカバリ戦略として再開またはスキップを使用できます。タスクが再開をリカバリ戦略として使用している場合は、タスクエラーが発生するとワークフローインスタンスが中断します。タスクがスキップをリカバリ戦略として使用している場合は、タスクエラーが発生するとデータ統合サービスがタスクをスキップします。

クラスタの作成タスク、マッピングタスク、クラスタの削除タスクを除くすべてのタスクが、再開をリカバリ戦略として使用します。クラスタの作成タスクは常にスキップをリカバリ戦略として使用します。マッピングタスクとクラスタの削除タスクは、リカバリ戦略として再開とスキップのどちらを使用するかを設定できます。

タスクでは次のリカバリ可能またはリカバリ不能なエラーが発生することがあります。

#### コマンドタスク

コマンドタスクエラーは、すべてリカバリ可能です。

#### ヒューマンタスク

ヒューマンタスクエラーはすべてリカバリ可能です。

**注:** ヒューマンタスクの `exceptionLoadCount` 変数値が 1 未満の場合、ヒューマンタスクは確認のためのタスクデータを生成しません。値が 1 未満の `exceptionLoadCount` 変数はタスクエラーを表しません。

#### マッピングタスク

マッピングタスク設定プロパティに割り当てられたワークフロー変数またはパラメータに無効な値が含まれる場合、マッピングタスクにリカバリ不能なエラーが発生します。例えば、値が「8」であるワークフローの文字列パラメータが最適化レベル設定プロパティに割り当てられた場合などです。

他のすべてのマッピングタスクエラーは、リカバリ可能です。

#### 通知タスク

通知タスクでは、次の状況でリカバリ可能なエラーが発生します。

- 電子メールサービスが利用できない場合。
- 電子メールサーバープロパティが正しく設定されていない場合。

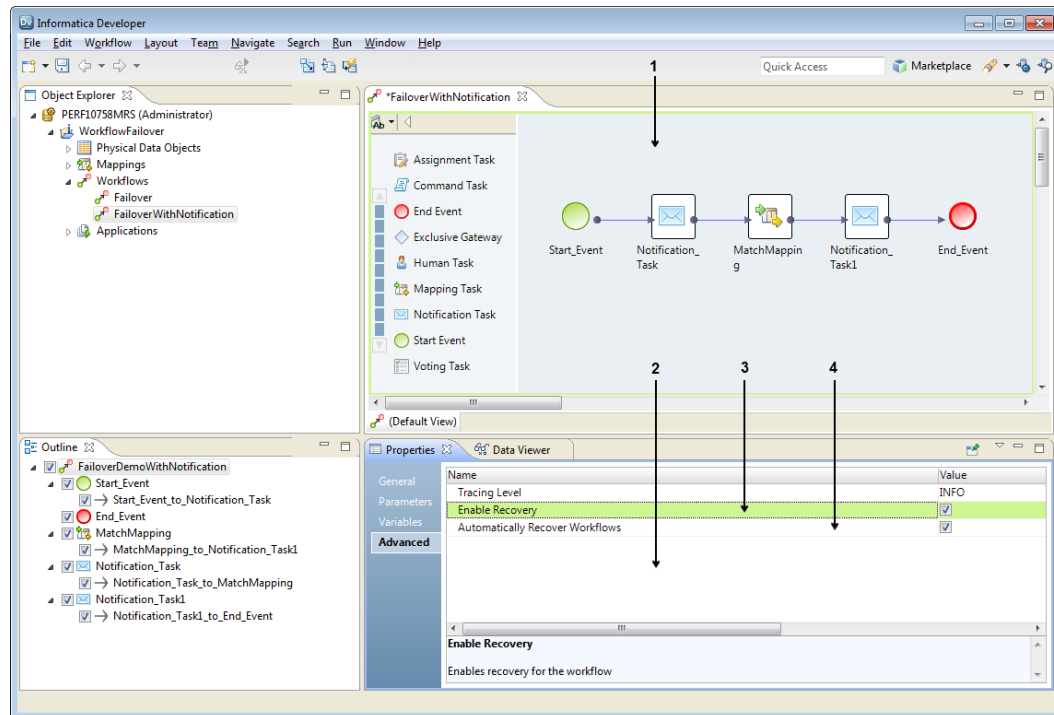
他のすべての通知タスクエラーは、リカバリ不能です。

## ワークフローリカバリストラテジ

ワークフローのリカバリを有効にする場合、リカバリ可能なエラーで中断されたワークフローのインスタンスに対して自動リカバリを設定できます。

自動リカバリに設定すると、データ統合サービスは、リカバリ可能なイベント発生時に、定義されたリカバリストラテジを適用します。サービスプロセスが再開されると、データ統合サービスプロセスはワークフローインスタンスをリカバリします。

次の図は、Developer tool のワークフローリカバリオプションを示します。



ワークフローリカバリストラテジを定義する場合、次のオプションを使用します。

1. Developer tool のワークフローエディタ。  
設定するワークフローがエディタに表示されます。
2. [プロパティ] ビューの [詳細] タブ。  
タブにはワークフローリカバリオプションが含まれます。
3. リカバリの有効化オプション。  
リカバリを有効にするには、このオプションを選択します。このオプションを選択すると、ワークフローインスタンスの自動リカバリオプションを選択またはクリアできます。
4. ワークフローの自動リカバリオプション。  
データ統合サービスによるワークフローのインスタンスの自動リカバリを有効にするには、このオプションを選択します。

## ワークフローリカバリと終結イベント

終結イベントで終了したワークフローはリカバリできません。アクティブなシーケンスフローが終結イベントに到達した場合、ワークフローは強制終了状態に入ります。

## グリッドでのワークフローリカバリ

グリッドで、あるノードから別のノードにフェイルオーバーするワークフローのステータスは、ノードの失敗時に実行されているタスクタイプにより異なります。

指定したワークフローリカバリストラテジに関係なく、実行中のワークフローではフェイルオーバー中に実行の継続を試みます。フェイルオーバーの時点で実行中のタスクがコマンドタスク、またはマッピングタスクの場合、ワークフローリカバリストラテジでワークフローの動作を決定します。



ワークフローに自動リカバリストラテジを指定すると、ワークフローではマッピングタスクまたはコマンドタスクが、その開始から再実行されます。ワークフローに自動リカバリストラテジを指定しないと、ワークフローはキャンセルされるので、アクティブなノード上で手動でワークフローをリカバリする必要があります。

データ統合サービスグリッドで実行されるワークフローに自動リカバリを設定できます。マスタサービスプロセスが予期せずシャットダウンした場合、マスタロールは別のサービスプロセスにフェイルオーバーします。フェイルオーバー中に実行されていて自動リカバリに設定されているワークフローインスタンスの場合、現在のマスタは自動でリカバリできます。

**注:** ワークフローにヒューマンタスクが含まれる場合は、ワークフローを自動リカバリに設定しないでください。ヒューマンタスクの前に実行されるマッピングタスク中にワークフローが中断されると、リカバリされたワークフローはマッピングタスクを再実行し、競合するデータをターゲットデータベースに追加します。

## タスクリカバリストラテジ

中断された、またはエラーが発生したタスクをデータ統合サービスが再実行するかスキップするかは、タスクリカバリストラテジによって決定されます。ワークフローのマッピングタスクのリカバリストラテジを確認します。

タスクには次のリカバリストラテジのいずれかを設定できます。

### リスタート

再開リカバリストラテジが設定されたタスクが中断されるか、タスクにリカバリ可能なエラーが発生した場合、データ統合サービスはタスクとワークフローをキャンセルします。データ統合サービスは、ワークフローのリカバリ時にタスクを再開します。再開リカバリストラテジが設定されたタスクでリカバリ不可能なエラーが発生した場合、データ統合サービスはタスクおよびワークフローを強制終了します。

ワークフローでリカバリを有効にすると、すべてのタスクではデフォルトで再開リカバリストラテジが使用されます。

ヒューマンタスクのデータを書き込むマッピングタスクに対して再開ストラテジを定義します。

### スキップ

スキップリカバリストラテジが設定されたタスクでリカバリ可能またはリカバリ不可能なエラーが発生した場合、データ統合サービスはそのタスクをスキップします。データ統合サービスはワークフローの次のステージを実行します。ワークフローでエラーや中断が発生しない場合、ワークフローは正常に完了します。

ワークフローでリカバリを有効にしていないと、データ統合サービスはタスクエラーが発生したタスクをスキップします。

## リカバリ動作の再開

再開リカバリストラテジが設定されているタスクが中断されると、タスクの状態は [キャンセル済み]、[強制終了]、または [完了] になります。データ統合サービスはキャンセル済みのタスクを再開できます。

次の表に、再開リカバリストラテジが設定されたタスクが中断された場合の、タスクおよびワークフローインスタンスの状態を示します。

中断	タスクの状態	ワークフローの状態	説明
タスクでリカバリ可能なエラーが発生	強制終了	強制終了	タスクとワークフローは強制終了されます。ワークフローはリカバリ不能です。
タスクでリカバリ可能なエラーが発生	キャンセル済み	キャンセル済み	タスクとワークフローはキャンセルされます。ワークフローはリカバリ可能です。
ユーザーがワークフローインスタンスを強制終了	強制終了	強制終了	データ統合サービスはタスクとワークフローインスタンスを強制終了します。ワークフローはリカバリ不能です。
ユーザーがワークフローインスタンスをキャンセル	完了/実行中	キャンセル済み	<p>データ統合サービスはワークフローインスタンスをキャンセルし、現在のタスクの実行を終了します。ワークフローはリカバリ可能です。</p> <p>現在のタスクは完了まで実行されるため、タスクがまだ実行中にワークフローステータスが [キャンセル済み] に変化する場合があります。</p> <p>タスクが完了すると、タスクがヒューマンタスクでない限り、[完了] 状態になります。ヒューマンタスクの場合は、ワークフローがリカバリするまで [実行中] 状態のままです。ワークフローがリカバリすると、ヒューマンタスクの状態は、タスクのステップの完了時に [完了] に変わります。</p>
サービスプロセスが予期せぬシャットダウン	キャンセル済み	キャンセル済み	<p>タスクとワークフローはキャンセルされます。ワークフローはリカバリ可能です。</p> <p>ワークフローで自動リカバリが設定されていない場合は、サービスプロセスの再開時に、タスクおよびワークフローインスタンスの状態が [キャンセル済み] に変更されます。ワークフローがリカバリされると、中断されたタスクは再開します。</p> <p>ワークフローで自動リカバリが設定されている場合は、サービスプロセスの再開時に、ワークフローインスタンスはリカバリされ、中断されたタスクは再開されます。サービスプロセスでタスクおよびワークフローインスタンスの状態が [実行中] に変更されます。</p> <p><b>注:</b> データ統合サービスの再開前に、ワークフローの状態とタスクの状態が [実行中] と表示されますが、ワークフローとタスクは実行中ではありません。データ統合サービスの再開時にワークフローおよびタスクの状態が更新されます。</p>

## スキップリカバリの動作

スキップリカバリストラテジが設定されているタスクが中断されると、タスクの状態は [失敗]、[完了]、[キャンセル済み]、または [強制終了] になります。ワークフローインスタンスがリカバリモードで実行されている場合、データ統合サービスは失敗および完了したタスクをスキップします。

次の表に、スキップリカバリストラテジが設定されたタスクが中断された場合の、タスクおよびワークフローインスタンスの状態を示します。

中断	タスクの状態	ワークフローの状態	説明
タスクでリカバリ可能またはリカバリ不能なエラーが発生した	失敗	完了	データ統合サービスがタスクを失敗にします。条件付きシーケンスフローの式が <code>true</code> に評価された場合、またはシーケンスフローに条件が含まれていない場合は、データ統合サービスで後続のワークフローオブジェクトが実行されます。他に中断せずにワークフローインスタンスの実行が終了した場合、データ統合サービスはワークフローの状態を [完了] に更新します。
ユーザーがワークフローインスタンスを強制終了	強制終了	強制終了	データ統合サービスはタスクを失敗にし、ワークフローを強制終了します。ワークフローはリカバリ不能です。
ユーザーがワークフローインスタンスをキャンセル	完了/実行中	キャンセル済み	データ統合サービスはワークフローインスタンスをキャンセルし、現在のタスクの実行を終了します。ワークフローはリカバリ可能です。 現在のタスクは完了まで実行されるため、タスクがまだ実行中にワークフローステータスが [キャンセル済み] に変化する場合があります。 タスクが完了すると、タスクがヒューマンタスクでない限り、[完了] 状態になります。ヒューマンタスクの場合は、ワークフローがリカバリするまで [実行中] 状態のままです。ワークフローがリカバリすると、ヒューマンタスクの状態は、タスクのステップの完了時に [完了] に変わります。
サービスプロセスが予期せぬシャットダウン	キャンセル済み	キャンセル済み	ワークフローとタスクはキャンセルされます。ワークフローはリカバリ可能です。 ワークフローで自動リカバリが設定されていない場合は、サービスプロセスの再開時に、ワークフローインスタンスの状態が [キャンセル済み] に変更されます。ワークフローをリカバリすると、サービスプロセスは、リカバリスキップストラテジに関係なく、中断されたタスクを再実行します。 ワークフローで自動リカバリが設定されている場合は、サービスプロセスの再開時に、ワークフローインスタンスはリカバリされ、中断されたタスクは再実行されます。サービスプロセスでワークフローインスタンスの状態が [実行中] に設定されます。リカバリスキップストラテジに関係なく、サービスプロセスにより、中断されたタスクが再開されます。 <b>注:</b> データ統合サービスの再開前に、ワークフローの状態とタスクの状態が [実行中] と表示されますが、ワークフローとタスクは実行中ではありません。データ統合サービスの再開時にワークフローおよびタスクの状態が更新されます。

## ヒューマンタスクの再開動作

ヒューマンタスクには再開リカバリストラテジが設定されています。他のタスクタイプと異なり、中断されたヒューマンタスクが再開される場合、操作を停止したポイントから続行されます。データ統合サービスは、ヒューマンタスクを最初から再開しません。

ヒューマンタスク動作に関する、次のルールおよびガイドラインを考慮してください。

- データ統合サービスがヒューマンタスクを実行すると、サービスはタスクインスタンスを作成し、Informatica ドメインのユーザーとグループに配布します。各タスクインスタンスは、ユーザーが Analyst ツールで分析する必要のあるワークフローデータのサブセットを識別します。

データ統合サービスが、ヒューマンタスクの中断前にすべてのタスクインスタンスを配布していない場合、サービスはタスクの再開時にタスクインスタンスの配布を続行します。ヒューマンタスクは最初から再開されません。

- データ統合サービスがヒューマンタスクの中断前にすべてのタスクインスタンスを配布した場合、Analyst ツールのユーザーはそのインスタンスで作業できます。ただし、ヒューマンタスクに複数のステップが含まれる場合、タスクデータは現在のタスクインスタンスから別のステップで定義されたタスクインスタンスへ移動できません。ワークフローが再開すると、ヒューマンタスクは中断したポイントから続行され、タスクデータは次のステップへ移動できます。

- ワークフローデータベースの接続がタスクインスタンスの作成中に失敗した場合、ワークフローは複数の異なる状態のうちの 1 つになります。ワークフローが失敗するか、データ統合サービスがヒューマンタスクで指定されたタスクインスタンスの一部を作成しない場合があります。

データ統合サービスがヒューマンタスクインスタンスのすべてを作成する前に、ワークフローデータベースの接続が失敗した場合は、データ統合サービスを停止します。ワークフローデータベースの接続が使用できることを確認し、データ統合サービスをリスタートします。ワークフローがリカバリすると、データ統合サービスは未配布になっているタスクインスタンスの配布を試行します。

- ワークフローが強制終了状態になると、ワークフローのヒューマンタスクは [完了] 状態になります。

**注:** ヒューマンタスクを含むワークフローをキャンセルし、アプリケーションの再デプロイを試みると、Developer tool は応答を停止することがあります。これは、ヒューマンタスクがタスク作成フェーズに属している場合に、ワークフローを強制終了すると発生することがあります。ヒューマンタスクからワークフローに制御が返されると、Developer tool は応答し、アプリケーションをデプロイします。

## ワークフローおよびタスクリカバリのルールとガイドライン

定義したワークフローの状態とタスクストラテジは、ワークフローインスタンスのリカバリ動作に影響を与えます。

ワークフローとタスクリカバリに関する次のルールとガイドラインに従ってください。

- スキップリカバリストラテジを設定したタスクにエラーが発生すると、データ統合サービスでタスクが失敗し、ワークフローの次のオブジェクトに移動します。ワークフローは中断されません。

スキップリカバリストラテジを設定したタスクの実行中にデータ統合サービスが中断された場合、ワークフローとタスクはキャンセルされた状態になります。ワークフローがリカバリされると、データ統合サービスでキャンセルされたタスクが再実行されます。

- 再開リカバリストラテジを設定したタスクでリカバリ可能なエラーが発生すると、データ統合サービスはワークフローをキャンセルします。ワークフローをリカバリすると、データ統合サービスはエラー発生時に実行していたタスクを再開します。

- スキップリカバリストラテジを設定したタスクを実行中にワークフローをキャンセルすると、データ統合サービスはワークフローをキャンセルし、タスクが終了するまで待機します。リカバリ時に、データ統合サービスはワークフローの次のオブジェクトを実行します。
- スキップリカバリストラテジを設定したタスクの実行中にワークフローが「強制終了」状態になると、データ統合サービスはタスクとワークフローを終了します。ワークフローはリカバリ不能です。
- タスクを実行するデータ統合サービスがグリッドまたは高可用性環境で失敗すると、別のデータ統合サービスでワークフローをリカバリできます。ワークフローエンジンが前のワークフロー実行からのタスクの最終状態を登録する前にサービスのフェイルオーバーが発生する場合があります。ワークフローをリカバリすると、データ統合サービスは前のワークフローの実行で完了した 1 つ以上のタスクを再実行する場合があります。

## リカバリの設定手順

リカバリを設定すると、中断ポイントからワークフローインスタンスをリカバリできます。

リカバリを設定するには、以下のタスクを実行します。

1. ワークフローのリカバリを設定します。
2. ワークフローの各マッピングタスクにリカバリストラテジを設定します。

**注:** 再開リカバリストラテジが設定された各マッピングタスクに、タスクの完全な再開をサポートするマッピングを開発します。

## ワークフローのリカバリの設定

ワークフローのリカバリを設定すると、リカバリ可能なエラーからワークフローインスタンスをリカバリできます。

1. エディタでワークフローを開きます。
2. **【プロパティ】** ビューで **【詳細】** タブをクリックします。
3. **【リカバリの有効化】** を選択します。
4. ワークフローのインスタンスの自動リカバリを設定するには、**【ワークフローの自動リカバリ】** を選択します。
5. **【ファイル】** > **【保存】** の順にクリックして、ワークフローを保存します。

## タスクリカバリストラテジの設定

デフォルトでは、すべてのタスクは再開リカバリストラテジを使用します。リカバリを有効にしたワークフローのマッピングタスクのタスクリカバリストラテジを確認します。

ワークフローでリカバリが有効になっていない場合は、データ統合サービスはタスクリカバリストラテジを無視します。

1. エディタでワークフローを開きます。
2. マッピングタスクを選択します。
3. タスクの **【プロパティ】** ビューで **【詳細】** タブをクリックします。
4. 次のいずれかのタスクリカバリストラテジを選択します。

- **タスクの再開。**再開リカバリストラテジが設定されたタスクが中断されるか、タスクにリカバリ可能なエラーが発生した場合、データ統合サービスはタスクとワークフローをキャンセルします。データ統合サービスは、ワークフローのリカバリ時にタスクを再開します。

ヒューマンタスクのデータを書き込むマッピングタスクに対して再開ストラテジを定義します。再開リカバリストラテジを使用するマッピングタスクを設定する場合、タスクの完全な再開をサポートするマッピングを開発します。

- **タスクのスキップ。**スキップリカバリストラテジが設定されたタスクでリカバリ可能またはリカバリ不能なエラーが発生した場合、データ統合サービスはそのタスクをスキップします。データ統合サービスはワークフローの次のステージを実行します。

5. ワークフローの他のマッピングタスクの手順を繰り返します。

6. **【ファイル】 > 【保存】** の順にクリックして、ワークフローを保存します。

## ワークフローリカバリ中のパラメータと変数の値

データ統合サービスは、ワークフローデータベースのワークフローパラメータおよび変数の現在の値を保存します。リカバリ済みのワークフローを実行する場合、データ統合サービスはワークフローの中断ポイントのところで、前のワークフロー実行で使用されたパラメータおよび変数を再利用します。

パラメータおよび変数には次の特性があります。

### ワークフローパラメータ

ワークフローパラメータでは、元のワークフローインスタンスの実行中にパラメータが取っていた値が使用されます。別のパラメータ値を使用してワークフローインスタンスをリカバリすることはできません。パラメータファイルの別の値を使用するには、`infacmd wfs startWorkflow` コマンドを使用して、ワークフローの他のインスタンスを実行する必要があります。

### ユーザー定義のワークフロー変数

ユーザー定義のワークフロー変数では、中断されたタスクを実行開始する前に変数が保持していた値が使用されます。

### システムワークフロー変数

`StartTime` および `InstanceId` システムワークフロー変数では、元のワークフローインスタンス実行中に変数が取っていた値が使用されます。

`UserName` システムワークフロー変数の値は、ユーザーがワークフローインスタンスをリカバリするか、データ統合サービスがワークフローインスタンスをリカバリするかによって異なります。ユーザーがワークフローインスタンスをリカバリする場合、`UserName` システムワークフロー変数はワークフローインスタンスをリカバリするユーザーの名前を使用します。データ統合サービスがワークフローインスタンスをリカバリする場合、`UserName` システムワークフロー変数は引き続き、前のワークフローインスタンスを実行したユーザーの名前を使用します。



# ワークフローリカバリログ

ワークフローインスタンスをリカバリする場合、データ統合サービスによって現在のワークフローログにログイベントが追加されます。リカバリされたワークフローインスタンスに再開されたマッピングタスクが含まれる場合、データ統合サービスはマッピングログを作成します。

ワークフローインスタンスがデータ統合サービスグリッド上で実行されている場合、ワークフローインスタンスのリカバリは、元のワークフローインスタンスとは別のノードで実行される可能性があります。リカバリが別のノードで実行され、ログディレクトリが共有ではない場合は、データ統合サービスにより現在のノード上に同じ名前で作成されたログファイルが作成されます。

## 再開のマッピングの開発手順

ターゲット、Java トランスフォーメーション、SQL トランスフォーメーションなどのマッピングオブジェクトは外部ファイルやデータベーステーブルに影響を与えることがあります。複数回マッピングを実行するとき、これらのマッピングオブジェクトにより、外部ファイルまたはデータベーステーブルで予想外の結果が生じることがあります。前のマッピングの実行から外部の影響を削除した後、リカバリ済みのワークフローでマッピングを再開するように、マッピングを開発する必要があります。

例えば、再開リカバリストラテジが設定されたマッピングタスクの実行中に、データ統合サービスが予期せず停止したとします。データ統合サービスが停止する前に、マッピングはターゲットに 50 行を書き込んでいます。ワークフローインスタンスをリカバリすると、マッピングタスクでは、中断ポイントでリカバリして行番号 51 の書き込みを開始するのではなく、マッピングが再開されます。

データ統合サービスが中断されたマッピングを再開する前に、前のマッピングがターゲットに書き込んだ行を手動で削除する必要があります。または、ターゲット行を削除するようにマッピングを設定します。

**注:** ターゲット行を手動で削除する場合は、ワークフローを自動リカバリに設定しないでください。

## 手動でのターゲット行の削除

保持する必要のあるデータを含む共有テーブルにマッピングが書き込む場合、元の実行で書き込まれた行を手動で削除してからワークフローインスタンスをリカバリできます。

ヒューマンタスクが読む取るテーブルをマッピングが書き込む場合は、ターゲット行を手動で削除する必要があります。複数のヒューマンタスクが同じデータベーステーブルにデータを書き込むことができます。マッピングを設定してターゲット行を削除する場合は、複数のマッピングからデータを消去する可能性があります。元のワークフロー実行時に書き込まれた行を識別して手動で削除するため、ワークフローインスタンス ID を使用します。Monitoring ツールのワークフロープロパティでワークフローインスタンス ID を確認できます。または、`infacmd wfs startWorkflow` コマンドおよび `listActiveWorkflowInstances` コマンドの出力でワークフローインスタンス ID を検索できます。

## ターゲット行を削除するマッピングの設定

保持する必要のあるデータを含まないファイルやテーブルにマッピングが書き込む場合は、すべてのデータをターゲットファイルまたはテーブルから削除するマッピングを設定できます。ヒューマンタスクが読み取らない共有テーブルにマッピングが書き込む場合は、元の実行で書き込まれた行を削除する SQL コマンドを実行するマッピングを設定できます。

次のいずれかの方法を使用してターゲット行を削除するように、マッピングを設定します。

#### フラットファイルターゲットを切り詰める。

ターゲットファイルからすべてのデータを削除できる場合は、ファイルに書き込む前にマッピングを設定して、ターゲットファイルを切り詰めます。

#### リレーショナルターゲットテーブルを切り詰める。

ターゲットテーブルからすべてのデータを削除できる場合は、データをロードする前にマッピングを設定して、ターゲットを切り詰めます。

#### リレーショナルターゲットテーブルで SQL コマンドを実行する。

ターゲットテーブルにあるデータの一部を保持する必要がある場合は、データ統合サービスを実行する SQL コマンドを作成して、ソースを読み取る前に中断されたマッピング実行で書き込まれた行を削除します。例えば、そのマッピングが複数部門で共有するテーブルに売り上げデータを書き込む場合は、営業部門用に書き込まれたすべての行を削除する SQL コマンドを作成します。または、そのマッピングが 1 日 1 回実行される場合は、現在の日付で書き込まれたすべての行を削除する SQL コマンドを作成します。

### フラットファイルターゲットを切り詰めるマッピングの構成

フラットファイルターゲットへの書き込みを行うマッピングの完全な再開をサポートするには、書き込みプロパティを設定して、ファイルに書き込む前にターゲットファイルを切り詰めます。

1. エディタでフラットファイルデータオブジェクトを開きます。
2. **【書き込み】** ビューを選択します。
3. Input トランスフォーメーションを選択します。
4. **【プロパティ】** ビューで **【ランタイム】** タブをクリックします。
5. **【存在する場合は追加】** をクリアします。
6. **【ファイル】** > **【保存】** の順にクリックし、フラットファイルデータオブジェクトを保存します。

### リレーショナルターゲットを切り詰めるマッピングの構成

リレーショナルターゲットへの書き込みを行うマッピングの完全な再開をサポートするには、マッピングを設定して、データをロードする前にターゲットを切り詰めます。

1. エディタでマッピングタスクを選択します。
2. **【プロパティ】** ビューで **【全般】** タブをクリックします。
3. タスクを実行するマッピングの名前をクリックします。  
マッピングが開きます。
4. エディタでリレーショナルデータオブジェクトを選択します。
5. **【プロパティ】** ビューで **【詳細】** タブをクリックします。
6. **【ターゲットテーブルの切り詰め】** を選択します。
7. **【ファイル】** > **【保存】** をクリックし、マッピングを保存します。

### SQL コマンドを実行するマッピングの設定

リレーショナルターゲットに書き込むマッピングの完全な再開をサポートするには、Data Integration Service がソースを読み取る前にターゲット行を削除するために実行する SQL コマンドを作成します。

1. エディタでマッピングタスクを選択します。
2. **【プロパティ】** ビューで **【全般】** タブをクリックします。
3. タスクを実行するマッピングの名前をクリックします。



マッピングが開きます。

4. エディタでリレーショナルデータオブジェクトを選択します。
5. **【プロパティ】** ビューで **【詳細】** タブをクリックします。
6. **【PreSQL】** プロパティの **【値】** カラムで、**【開く】** ボタンをクリックします。  
**【SQL クエリ】** エディタが表示されます。
7. SQL コマンドを入力し、**【OK】** をクリックします。  
Developer ツールでは、SQL は検証されません。
8. **【ファイル】** > **【保存】** をクリックし、マッピングを保存します。

## ワークフローインスタンスのリカバリ

キャンセルされたワークフローインスタンスをリカバリするには、Monitoring ツールを使用するか、`infacmd wfs recoverWorkflow` コマンドを実行します。

Monitoring ツールには、実行中、完了、および中断されたワークフローインスタンスのステータスが表示されます。Monitoring ツールを使用して、中断されたワークフローインスタンスのログを確認し、中断の原因を調査します。リカバリ可能なエラーを修正後、リカバリが有効になっている場合は中断されたワークフローインスタンスをリカバリできます。

コマンドラインからワークフローインスタンスをリカバリするには、ワークフローインスタンス ID を指定する必要があります。ワークフローインスタンス ID は、Monitoring ツールのワークフロープロパティで確認できます。または、`infacmd wfs startWorkflow` コマンドおよび `infacmd wfs listActiveWorkflowInstances` コマンドの出力でワークフローインスタンス ID を確認できます。

例えば、以下のコマンドは、ワークフローインスタンス ID が `u-6-j4MwEeGxHO9AUHdw6A` である中断されたワークフローインスタンスをリカバリします。

```
infacmd wfs recoverWorkflow -dn MyDomain -sn MyDataIntSvs -un MyUser -pd MyPassword -iid u-6-j4MwEeGxHO9AUHdw6A
```

## 中断後のワークフローの状態のサマリ

ワークフローインスタンスが中断されときのワークフローの状態とタスクの状態は、中断の原因および選択したリカバリオプションにより異なります。

**リカバリが有効でない場合。**

次の表は、ワークフローのリカバリが有効でない場合のワークフローの状態とタスクの状態のサマリです。

中断	タスクの状態	ワークフローの状態
ワークフローまたはタスクでエラーが発生	失敗	完了
ユーザーがワークフローインスタンスを強制終了	強制終了	強制終了
ユーザーがワークフローインスタンスをキャンセル	完了	キャンセル済み

リカバリが有効で、中断されたタスクには再開リカバリストラテジが設定されている場合。

次の表は、ワークフローのリカバリが有効であり、中断されたタスクに再開リカバリストラテジが設定されている場合のワークフローの状態とタスクの状態のサマリです。

中断	タスクの状態	ワークフローの状態
ワークフローでエラーが発生	強制終了	強制終了
タスクでリカバリ不能なエラーが発生	強制終了	強制終了
タスクでリカバリ可能なエラーが発生	キャンセル済み	キャンセル済み
ユーザーがワークフローインスタンスを強制終了	強制終了	強制終了
ユーザーがワークフローインスタンスをキャンセル	完了	キャンセル済み

リカバリが有効で、中断されたタスクにスキップリカバリストラテジが設定されている場合。

次の表は、ワークフローのリカバリが有効であり、中断されたタスクにスキップリカバリストラテジが設定されている場合のワークフローの状態とタスクの状態のサマリです。

中断	タスクの状態	ワークフローの状態
ワークフローでエラーが発生	強制終了	強制終了
タスクにエラーが発生	失敗	完了
ユーザーがワークフローインスタンスを強制終了	強制終了	強制終了
ユーザーがワークフローインスタンスをキャンセル	完了	キャンセル済み

## ワークフローリカバリの例

次の例では、中断されたワークフローインスタンスについて、およびデータ統合サービスが各インスタンスのリカバリを試行する方法について説明します。各々の例では、ワークフローのリカバリは有効になっており、各タスクはコマンドタスク、マッピングタスク、または通知タスクです。

ワークフローで次の中断が発生します。

**ユーザーがワークフローインスタンスをキャンセルする。**

タスク 2 の実行中に、ユーザーがワークフローインスタンスをキャンセルします。データ統合サービスはタスク 2 を完了してから、ワークフローインスタンスをキャンセルします。3 つのタスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (完了) ->タスク 2 (完了) ->タスク 3 (未開始)。ワークフローはキャンセルされます。

ワークフローをリカバリすると、データ統合サービスはタスク 1 とタスク 2 を無視し、タスク 3 を初めて実行します。

**スキップリカバリストラテジが設定されたタスクの実行中に、ユーザーがワークフローインスタンスを強制終了する。**

タスク 2 の実行中に、ワークフローインスタンスが強制終了されます。タスク 2 にはスキップリカバリストラテジが設定されています。データ統合サービスはタスク 2 を強制終了し、ワークフローインスタンスを強制終了します。3 つのタスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (完了) ->タスク 2 (強制終了) ->タスク 3 (未開始)。ワークフローは強制終了されます。

ワークフローを強制終了したため、ワークフローインスタンスはリカバリできません。

**データ統合サービスプロセスが予期せずにシャットダウンした。ワークフローは、リカバリに設定されている。**

タスク 2 の実行中、データ統合サービスプロセスがシャットダウンしました。タスク 2 は通知タスクです。サービスプロセスが再開したときに、3 つのタスクとワークフローインスタンスの状態は次のようになります。

タスク 1 (完了) ->タスク 2 (キャンセル) ->タスク 3 (未開始)。ワークフローはキャンセルされます。

ワークフローをリカバリすると、データ統合サービスはタスク 1 を無視し、タスク 2 を再開して、タスク 3 を初めて実行します。

**再開リカバリストラテジが設定されたタスクで、リカバリ可能なエラーが発生した。**

再開リカバリストラテジが設定されているタスク 2 で、リカバリ可能なエラーが発生しました。サービスはタスク 2 をキャンセルし、ワークフローインスタンスをキャンセルします。3 つのタスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (完了) ->タスク 2 (キャンセル) ->タスク 3 (未開始)。ワークフローはキャンセルされます。

ワークフローをリカバリすると、データ統合サービスはタスク 1 を無視し、タスク 2 を再開して、タスク 3 を初めて実行します。

**スキップリカバリストラテジが設定されたタスクでエラーが発生し、ユーザーがワークフローインスタンスをキャンセルする。**

タスク 2 にはスキップリカバリストラテジが設定されています。タスクでエラーが発生して失敗します。出力シーケンスフローには、タスク 2 が成功したことを確認する条件は含まれていません。このため、データ統合サービスは後続のワークフローオブジェクトを引き続き実行します。タスク 3 の実行中に、ワークフローインスタンスがキャンセルされます。タスク 3 には再開リカバリストラテジが設定されています。データ統合サービスはタスク 3 をキャンセルしてから、ワークフローインスタンスをキャンセルします。4 つのタスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (完了) ->タスク 2 (失敗) ->タスク 3 (キャンセル) ->タスク 4 (未開始)。ワークフローはキャンセルされます。

ワークフローをリカバリすると、データ統合サービスはタスク 1 とタスク 2 を無視し、タスク 3 を再開して、タスク 4 を初めて実行します。リカバリの実行中に、タスク 3 とタスク 4 は正常に完了します。データ統合サービスはワークフローの最終状態を [完了] に更新します。

**スキップリカバリストラテジが設定されたタスクでエラーが発生し、ワークフローインスタンスを強制終了した場合。**

タスク 2 にはスキップリカバリストラテジが設定されています。タスクでエラーが発生して失敗します。出力シーケンスフローには、タスク 2 が成功したことを確認する条件は含まれていません。このため、データ統合サービスは後続のワークフローオブジェクトを引き続き実行します。タスク 3 の実行中に、ワークフローインスタンスが強制終了されます。タスク 3 には再開リカバリストラテジが設定されています。データ統合サービスはタスク 3 を強制終了してから、ワークフローインスタンスを強制終了します。4 つのタスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (完了) ->タスク 2 (失敗) ->タスク 3 (強制終了) ->タスク 4 (未開始)。ワークフローは強制終了されます。

ワークフローを強制終了したため、ワークフローインスタンスはリカバリできません。

**スキップリカバリストラテジが設定されたタスクでエラーが発生し、出力条件付きシーケンスフローでタスクの失敗がチェックされる。**

タスク 1 にはスキップリカバリストラテジが設定されています。タスクでエラーが発生して失敗します。出力シーケンスフローには、タスク 1 が成功したことを確認する条件が含まれます。条件で false が返されたため、データ統合サービスは後続のワークフローオブジェクトの処理を停止し、ワークフローインスタンスを完了させます。3 つのタスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (失敗) ->タスク 2 (未開始) ->タスク 3 (未開始)。ワークフローは完了します。

ワークフローインスタンスが完了したため、ワークフローインスタンスのリカバリはできません。

**リカバリをサポートしていないワークフローでタスクエラーが発生したため、タスクが失敗する。**

ワークフローでワークフローのリカバリストラテジが指定されていないため、データ統合サービスはタスクエラーが原因で失敗するタスクをスキップします。ワークフローは完了まで実行されます。3 つのタスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (完了) >タスク 2 (失敗) >タスク 3 (完了)。ワークフローは完了します。

ワークフローインスタンスが完了したため、ワークフローインスタンスのリカバリはできません。

**マッピングタスクのマッピングによってヒューマンタスクの例外データが生成されない。ワークフローのアクティブなシーケンスフローが終結イベントに到達した。**

タスク 1 はマッピングタスクであり、タスク 2 は通知タスクです。マッピングタスクは、ダウンストリームヒューマンタスクの例外データを生成します。通知タスクは、マッピングタスクのマッピングによって生成される例外の行数が含まれる電子メールを送信します。ワークフローには、終結イベントとヒューマンタスクに接続する排他ゲートウェイが含まれます。

ゲートウェイの最初のシーケンスフローは終結イベントに接続し、マッピングタスクの出力を評価する条件が含まれます。ゲートウェイの 2 番目のシーケンスフローはヒューマンタスクに接続します。タスク 1 で指定されるマッピングが例外データを生成しないため、ゲートウェイの条件付きシーケンスフローは終結イベントをトリガします。タスクとワークフローインスタンスの状態は、次のようになります。

タスク 1 (完了) ->タスク 2 (完了) ->終結イベント。ワークフローは強制終了されます。

ワークフローを意図的に強制終了状態にしているため、ワークフローをリカバリすることはできません。

# 第 11 章

## ワークフロー管理

この章では、以下の項目について説明します。

- [ワークフロー管理の概要, 141 ページ](#)
- [ワークフローグラフ, 142 ページ](#)
- [ワークフローオブジェクトの表示, 142 ページ](#)
- [ワークフローの状態, 144 ページ](#)
- [ワークフローオブジェクトの状態, 145 ページ](#)
- [マッピングタスクの作業項目の状態, 146 ページ](#)
- [ワークフローのキャンセルまたは強制終了, 146 ページ](#)
- [ワークフローリカバリ, 147 ページ](#)
- [ワークフローログ, 148 ページ](#)

## ワークフロー管理の概要

データ統合サービスにワークフローをデプロイ後、Administrator ツールまたは Monitor ツールを使用してワークフロージョブの管理と監視ができます。これらのタスクを実行するには、適切な特権が必要です。

ワークフロージョブは、以下の場所で監視できます。

- Monitoring ツール。Developer tool で、**進行状況ビュー**の **【メニュー】** ボタンをクリックし、**【ジョブの監視】** を選択します。ワークフローを実行しているデータ統合サービスを選択し、**【OK】** をクリックします。Monitoring ツールが開きます。
- Administrator ツール。Administrator ツールでワークフロージョブを監視するには、**【モニタ】** タブをクリックします。

ワークフロージョブを監視する場合、ジョブのサマリ統計または実行統計を表示できます。**【サマリ統計】** ビューには、ドメイン内におけるワークフロージョブのステータスの概要が図で表示されます。

**【実行統計】** ビューには、ワークフロージョブとワークフローオブジェクトに関する情報が表示されます。

**【実行統計】** ビューの **【コンテンツ】** パネルでワークフロージョブを選択すると、次のタスクを実行できます。

- ワークフローのグラフの表示。
- ワークフローオブジェクトのプロパティの表示。
- ワークフローインスタンスのキャンセルまたは強制終了。
- 中断されたワークフローインスタンスのリカバリ。

- ワークフローインスタンスのログの表示。

## ワークフローグラフ

実行するワークフローの詳細を Monitoring ツールでグラフィック形式で表示できます。

ワークフローを実行した後、Monitoring ツールでワークフローのグラフィカルビューを表示できます。ワークフローグラフでは、ワークフロー内のマッピングタスクの連続実行を確認できます。ワークフローグラフを使用すると、ワークフロー内の障害ポイントを即座に確認できます。

ワークフローグラフでは、以下のワークフロー詳細を確認できます。

- ワークフロー内のマッピングタスク
- タスクの詳細
- リカバリの詳細

ワークフローグラフでは以下のタスクを実行できます。

- ワークフロー実行の強制終了
- 実行中のワークフローのキャンセル
- 停止したワークフローのリカバリ
- ワークフローログの表示

## ワークフローのグラフの表示

ワークフローのマッピングタスクのシーケンシャルな実行を示すワークフローのグラフを表示できます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、アプリケーションを展開します。
3. **【ワークフロー】** フォルダを選択します。  
[コンテンツ] パネルにワークフローのリストが表示されます。
4. 表示するワークフローを選択します。
5. **【アクション】** > **【ワークフローのグラフの表示】** をクリックします。  
新規ウィンドウにワークフローのグラフが表示されます。

## ワークフローオブジェクトの表示

[コンテンツ] パネルでワークフローインスタンスを展開すると、オブジェクトの名前、状態、開始時刻、経過時間などのワークフローオブジェクトに関するプロパティを表示できます。

ワークフローオブジェクトには、イベント、タスク、ゲートウェイが含まれます。ワークフローを監視する場合、ワークフローインスタンスで実行するタスクを監視できます。**【モニタ】** タブには、ワークフローインスタンスのイベントやゲートウェイに関する情報は表示されません。

条件付きシーケンスフローの式が false に評価される場合、データ統合サービスは次のオブジェクトまたはブランチ内の後続のオブジェクトを実行しません。**【モニタ】** タブには、ワークフローインスタンスで実行され

ないオブジェクトはリスト表示されません。ワークフローインスタンスに実行されないオブジェクトが含まれている場合でも、インスタンスは正常に完了します。

[コンテンツ] パネルでタスクを展開して、そのタスクによって実行された作業項目の情報を表示できます。例えば、ワークフローにマッピングタスクが含まれる場合、マッピング実行のスループットおよびリソース使用状況の統計を表示できます。

## ワークフローオブジェクトのサマリ統計の表示

別のローカルプロセスで実行されるワークフローのマッピングオブジェクトについて、スループットおよびリソース使用状況の統計を表示できます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. アプリケーションを展開して **【ワークフロー】** フォルダを選択します。  
[コンテンツ] パネルにワークフローのリストが表示されます。
4. マッピングオブジェクトを含むワークフローを展開します。
5. マッピングタスクを展開し、マッピングを選択します。
6. **【詳細】** パネルで、**【サマリ統計】** ビューをクリックします。  
**【サマリ統計】** ビューに、ソースおよびターゲットのスループットとリソース使用状況の統計が表示されます。

必要に応じて、昇順または降順で統計をソートすることができます。列ヘッダーをクリックして昇順に列をソートします。再度列ヘッダーをクリックして降順に列をソートします。

## ワークフローオブジェクトの詳細統計の表示

別のローカルプロセスで実行されるワークフローのマッピングオブジェクトについて、スループットおよびリソース使用状況のグラフを表示できます。1分を超えて実行されるジョブについての詳細統計が表示されます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. アプリケーションを展開して **【ワークフロー】** フォルダを選択します。  
[コンテンツ] パネルにワークフローのリストが表示されます。
4. マッピングオブジェクトを含むワークフローを展開します。
5. マッピングタスクを展開し、マッピングを選択します。
6. **【詳細】** パネルの **【詳細統計】** ビューをクリックします。  
**【詳細統計】** ビューにスループットグラフとリソース使用状況グラフが表示されます。

必要に応じて、**【詳細統計】** ビューで次のタスクを実行することもできます。

タスク	説明
グラフの拡大	グラフ上にカーソルを移動し、拡大鏡アイコンをクリックします。
拡大されたグラフの一部を拡大	カーソルをドラッグして拡大する領域を選択します。

タスク	説明
スループットグラフの行とバイトの切り替え	[バイト] オプションまたは [行] オプションをクリックします。
スループットグラフにプロットする統計の選択	[スループット] フィールドで、表示するソースとターゲットを選択します。

## ワークフローの状態

ワークフローインスタンスを監視する場合には、ワークフローインスタンスの状態を表示できます。タスクの中断後にワークフローインスタンスをリカバリする場合、モニタはリカバリされたワークフローで実行されるタスクインスタンスのエントリを追加します。

ワークフローインスタンスの状態は、次のいずれかになります。

### 強制終了

ワークフローインスタンスは、Monitoring ツールからワークフローインスタンスの強制終了を選択するか、`infacmd wfs abortWorkflow` コマンドを使用した場合に強制終了します。ワークフローを含むアプリケーションを停止するとき、またはアプリケーションのワークフローを無効化するときに、実行中のワークフローインスタンスを強制終了することもできます。

**注:** ワークフローインスタンスは、ワークフローのアクティブなシーケンスフローが終点イベントに到達した場合も強制終了します。

### キャンセル済み

**【モニタ】** タブから、または `infacmd wfs cancelWorkflow` コマンドを使用して、ワークフローインスタンスをキャンセルすることをユーザーが選択した場合。

データ統合サービスが予期せぬシャットダウンをした場合もワークフローは [キャンセル済み] 状態になります。ワークフローで自動リカバリが設定されていない場合は、サービスプロセスの再開時に、ワークフローインスタンスの状態が [キャンセル済み] に変更されます。データ統合サービスの再開前に、ワークフローの状態とタスクの状態が [実行中] と表示されますが、ワークフローとタスクは実行中ではありません。ワークフローで自動リカバリが設定されている場合は、サービスプロセスの再開時に、ワークフローインスタンスはリカバリされ、中断されたタスクは再実行されます。サービスプロセスでワークフローインスタンスの状態が [実行中] に設定されます。

### 完了

データ統合サービスはワークフローインスタンスを正常に完了します。完了したワークフローインスタンスとは、すべてのタスク、ゲートウェイ、およびシーケンスフローの評価が正常に完了したか、実行されなかったブランチに入っていたことを示す場合があります。

ワークフローは、コマンド、マッピング、通知、またはヒューマンタスクでリカバリ可能またはリカバリ不能なエラーが発生した場合も [完了] 状態になることがあります。タスクにエラーが発生すると、データ統合サービスはタスクを失敗にします。条件付きシーケンスフローの式が `true` に評価された場合、またはシーケンスフローに条件が含まれていない場合は、データ統合サービスで後続のワークフローオブジェクトが実行されます。他に中断せずにワークフローインスタンスの実行が終了した場合、データ統合サービスはワークフローの状態を [完了] に更新します。

タスクが失敗した場合、条件付きシーケンスフローの式が `true` に評価されるか、またはシーケンスフローに条件が含まれていないと、データ統合サービスはワークフローインスタンスの以降のオブジェクトを引



き続き実行します。他に中断せずにワークフローインスタンスの実行が終了した場合、データ統合サービスはワークフローの状態を「完了」に更新します。完了したワークフローインスタンスには、失敗したタスクと完了したタスクの両方が含まれている可能性があります。

#### 失敗

ワークフローエラーが発生すると、ワークフローインスタンスは失敗します。ワークフローエラーは、データ統合サービスがワークフロー実行の開始時にパラメータファイルを読み取るとき、ワークフローパラメータおよび変数の値をタスク入力にコピーするとき、または条件付きシーケンスフローで式を評価するときに発生することがあります。また、割り当てタスクまたはゲートウェイが失敗した場合にもワークフローエラーが発生します。

ワークフローエラーが発生すると、データ統合サービスは以降のオブジェクトの処理を停止し、ワークフローインスタンスは即座に失敗します。ワークフローエラーはリカバリ不能です。

#### 実行中

データ統合サービスはワークフローインスタンスを実行しています。

## ワークフローオブジェクトの状態

ワークフローにはタスクとゲートウェイが含まれます。ワークフローインスタンスを監視する場合には、ワークフローインスタンスで実行されるタスクの状態を表示できます。

タスクは次のいずれかの状態にあります。

#### 強制終了

タスクは、次の状況で強制終了します。

- タスクにリカバリ不能なエラーが発生した場合。
- ワークフローインスタンスを強制終了した場合。

ワークフローインスタンスを強制終了する場合、データ統合サービスはまずタスクを強制終了し、次にワークフローインスタンスを強制終了します。

割り当てタスクの実行中にワークフローインスタンスの強制終了を選択すると、データ統合サービスはそのタスクの実行を完了します。データ統合サービスは次にワークフローインスタンスを強制終了し、他のオブジェクトの実行を開始することはありません。

#### 完了

データ統合サービスはタスクを正常に完了します。

#### 失敗

タスクは、次の状況で失敗します。

- リカバリが有効になっていないワークフロー内のタスクで、何らかのタイプのエラーが発生した場合。
- リカバリが有効になっているワークフロー内の割り当てタスクで、何らかのタイプのエラーが発生した場合。
- 再開リカバリストラテジが設定された、リカバリが有効になっているワークフロー内のコマンド、マッピング、通知、またはヒューマンタスクで、リカバリ不能なエラーが発生した場合。
- スキップリカバリストラテジが設定された、リカバリが有効になっているワークフロー内のマッピングタスクで、何らかのタイプのエラーが発生した場合。

**注:** タスクが失敗してもワークフローが完了することがあります。条件付きシーケンスフローの式が true に評価された場合、またはシーケンスフローに条件が含まれていない場合は、データ統合サービスで後続のワークフローオブジェクトが実行されます。他に中断せずにワークフローインスタンスの実行が終了した場合、データ統合サービスはワークフローの状態を [完了] に更新します。

#### 実行中

データ統合サービスはタスクを実行中です。

## マッピングタスクの作業項目の状態

マッピングタスクを展開すると、マッピングの実行状態を表示できます。再開されたマッピングタスクを展開すると、ワークフローインスタンスのリカバリ試行ごとに実行したマッピングジョブを表示できます。マッピングタスクの中断後にワークフローインスタンスをリカバリする場合、モニタはリカバリされるワークフローで実行されるタスクインスタンスのエントリを追加します。

マッピング実行の状態は、マッピングタスクが含まれるワークフローのワークフローグラフからも確認できます。

マッピングタスクによって実行されたマッピングは、以下のいずれかの状態になります。

#### 強制終了

ユーザーがワークフローインスタンスの強制終了を選択したため、マッピングの実行中にマッピングタスクが強制終了されます。

#### 完了

データ統合サービスはマッピングを正常に完了します。

#### 失敗

マッピングでエラーが発生します。マッピングおよびマッピングタスクはモニタに [失敗] と表示されます。状態はマッピングタスクリカバリストラテジに依存しません。

#### 実行中

データ統合サービスはマッピングを実行中です。

## ワークフローのキャンセルまたは強制終了

ワークフローインスタンスをいつでもキャンセルまたは強制終了できます。ワークフローインスタンスが応答しなくなったり、または処理に必要以上に時間がかかっている場合、ワークフローインスタンスをキャンセルまたは強制終了することができます。

ワークフローインスタンスをキャンセルすると、データ統合サービスは実行中のすべてのタスクの処理を終了してからワークフローインスタンスの処理を停止します。以降のワークフローオブジェクトの実行が開始されることはありません。

ワークフローインスタンスを強制終了すると、データ統合サービスは実行中のすべてのタスクに対してプロセスの終了を試みます。割り当てタスクまたはゲートウェイが実行中の場合、データ統合サービスはそのタスクまたはゲートウェイを完了します。タスクが強制終了または完了した後、サービスはワークフローインスタンスを強制終了します。以降のワークフローオブジェクトの実行が開始されることはありません。

ワークフローグラフからワークフローをキャンセルまたは強制終了することも可能です。

1. **【実行統計】** ビューをクリックします。
2. ナビゲータで、データ統合サービスを展開します。
3. アプリケーションを展開し、**【ワークフロー】** を選択します。  
ワークフローインスタンスのリストが **【コンテンツ】** パネルに表示されます。
4. ワークフローインスタンスを選択します。
5. **【アクション】 > 【選択したワークフローのキャンセル】** または **【アクション】 > 【選択したワークフローの強制終了】** をクリックします。

## ワークフローリカバリ

ワークフローリカバリとは、中断ポイントからワークフローインスタンスを完了することです。

ワークフローでリカバリを有効にすると、タスクでリカバリ可能なエラーが発生した場合、ワークフローインスタンスをキャンセルした場合、またはデータ統合サービスプロセスが予期せずにシャットダウンした場合に、ワークフローインスタンスをリカバリできます。

ワークフローログを確認して、中断の原因を特定します。リカバリ可能なエラーを修正後、リカバリが有効になっている場合は中断されたワークフローインスタンスをリカバリできます。

中断された実行とリカバリ実行間のワークフロー定義を変更することはできません。ワークフローインスタンスがリカバリ可能な状態で、Developer tool のワークフローメタデータを変更し、ワークフローが含まれるアプリケーションを再デプロイすると、ワークフローインスタンスはリカバリ不能になります。

データ統合サービスは、予期しないシャットダウン後にサービスを再開する場合、前のワークフローの状態にリカバリしようとしています。デフォルトでは、データ統合サービスはコマンドタスク、マッピングタスク、または通知タスクの実行中に停止したワークフローインスタンスをリカバリしません。また、データ統合サービスは、ワークフローインスタンスまたはワークフローインスタンスで実行中のタスクをキャンセルした場合、デフォルトでは、ワークフローインスタンスをリカバリできません。ワークフローのリカバリオプションを設定して、データ統合サービスによる上記の場合のワークフローインスタンスのリカバリを有効にできます。

ワークフローオプションを設定する際に、ワークフローを手動リカバリまたは自動リカバリに設定できます。自動リカバリに設定すると、データ統合サービスは、人の操作なしに、中断ポイントからワークフローを再開します。手動リカバリに設定すると、手動でワークフローを再開できます。

ワークフローインスタンスが自動または手動でリカバリされると、データ統合サービスはタスクを再開します。サービスで後続のワークフローオブジェクトの処理が続行されます。タスクの中断後にワークフローインスタンスをリカバリする場合、モニタはリカバリされるワークフローで実行されるタスクインスタンスのエントリを追加します。例えば、ワークフローが3回リカバリされ、そのたびにマッピングタスクが再開されると、モニタにはマッピングタスクの3つのエントリが含まれます。

## リカバリプロパティ

各ワークフローインスタンスのリカバリのプロパティが読み取り専用で表示されます。Developer tool でワークフロー定義のリカバリプロパティを設定します。ワークフローインスタンスのプロパティの値を変更することはできません。

以下の表に、ワークフローインスタンスのリカバリのプロパティを読み取り専用で示します。

プロパティ	説明
リカバリが有効	ワークフローでリカバリが有効になっていることを示します。
ワークフローの自動リカバリ	データ統合サービスプロセスは、中断されたワークフローインスタンスの自動リカバリを試行することを示します。ワークフローリカバリは、データ統合サービスプロセスの再開後に開始されます。

## ワークフローのリカバリ

リカバリが有効になった中断されたワークフローインスタンスをリカバリできます。

1. **【実行統計】** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. アプリケーションを展開し、**【ワークフロー】** を選択します。  
ワークフローインスタンスのリストが **【コンテンツ】** パネルに表示されます。
4. リカバリする中断されたワークフローインスタンスを選択します。
5. **【アクション】** > **【選択したワークフローのリカバリ】** の順にクリックします。  
**【コンテンツ】** パネルでワークフローリカバリ実行の状態を監視します。

## ワークフローログ

ワークフローを実行すると、データ統合サービスはログイベントを生成します。ログイベントには、ワークフローエラー、タスク進捗度、およびワークフロー変数の設定についての情報が含まれます。また、データ統合サービスがシーケンスフロー内で評価するリンクの分析も含まれます。

ワークフローインスタンスにマッピングタスクが含まれる場合、マッピング用に別のログファイルが生成されます。マッピングログファイルには、マッピングの実行時、サマリおよびトランスフォーメーション統計のロード時に発生したエラーが含まれます。

ワークフローログおよびマッピングログは、**【モニタ】** タブから表示できます。

中断されたワークフローインスタンスをリカバリすると、現在のワークフローログにログイベントが追加されます。リカバリされたワークフローインスタンスに再開されたマッピングタスクが含まれる場合は、マッピングログが作成されます。

ワークフローがグリッド上で実行されている場合、ワークフローインスタンスのリカバリは、元のワークフローインスタンスとは別のノードで実行される可能性があります。リカバリが別のノードで実行され、ログディレクトリが共有されていない場合は、現在のノード上に同じ名前で作成されます。

## ワークフローログ情報

ワークフローログファイル内の情報は、そのワークフローの実行時に発生するイベントのシーケンスを表しています。

データ統合サービスは、次のタイプのイベントが発生したときに、ワークフローログに情報を書き込みます。

- データ統合サービスがワークフロー内のタスクまたは別のオブジェクトの実行を開始したとき。
- ワークフロー内のタスクまたは別のオブジェクトが実行中のとき。
- データ統合サービスが、ワークフロー内のタスクまたは別のオブジェクトの実行を終了したとき。
- データ統合サービスがワークフロー変数を設定または更新したとき。
- データ統合サービスが、シーケンスフロー内のリンクを評価して、ワークフロープロセスの正しいパスを特定したとき。
- ワークフローにエラーが発生したとき。

## ワークフローのログ表示

ワークフローインスタンスのログをダウンロードして、ワークフローインスタンスの詳細を表示できます。

1. Administrator ツールで、**[モニタ]** タブをクリックします。
2. **[実行統計]** ビューをクリックします。
3. ドメインナビゲータで、データ統合サービスを展開します。
4. アプリケーションを展開し、**[ワークフロー]** を選択します。  
ワークフローインスタンスのリストが **[コンテンツ]** パネルに表示されます。
5. ワークフローインスタンスを選択します。
6. **[アクション]** > **[選択したオブジェクトのログの表示]** をクリックします。  
ログファイルを開く、または保存するオプションのついたダイアログボックスが表示されます。

## ワークフロー内のマッピング実行のログ表示

ワークフローでのマッピング実行のログをダウンロードして、マッピングの詳細を表示できます。

1. **[実行統計]** ビューをクリックします。
2. ドメインナビゲータで、データ統合サービスを展開します。
3. アプリケーションを展開し、**[ワークフロー]** を選択します。  
ワークフローインスタンスのリストが **[コンテンツ]** パネルに表示されます。
4. ワークフローインスタンスを展開します。
5. マッピングタスクを展開して、そのタスクによって実行されたマッピングを選択します。
6. **[アクション]** > **[選択したオブジェクトのログの表示]** をクリックします。  
ログファイルを開く、または保存するオプションのついたダイアログボックスが表示されます。

# 索引

## D

Databricks

ウォームプール、設定 [56](#), [58](#)  
クラスタの作成タスク [56](#), [58](#)

## L

listMappingPersistedOutputs  
説明 [92](#)

## U

updateMappingPersistedOutputs  
説明 [92](#)

## あ

アクティビティ  
概要 [13](#)

## い

イベント

開始イベント [13](#)  
終結イベント [13](#)  
終了イベント [13](#)  
ワークフローへの追加 [17](#)

## う

ウォームプール [56](#), [58](#)

## え

エスケープ文字

ワークフローパラメータ [41](#)  
ワークフロー変数 [34](#)

## お

オブジェクト

ワークフローへの追加 [17](#)

## か

開始イベント [13](#)

カスタムプロパティ  
マスキングタスクの設定 [107](#)

## く

クラスタタスク

概要 [52](#)

クラスタワークフロー

クラスタワークフロー内のマッピングタスク [110](#)

グリッド

自動ワークフローリカバリ [127](#)

## け

ゲートウェイ

概要 [14](#), [119](#)

条件付きシーケンスフロー [123](#)

ネストされたゲートウェイ [119](#)

排他ゲートウェイ [120](#)

包含ゲートウェイ [121](#)

ワークフローの分割 [123](#)

ワークフローのマージ [124](#)

ワークフローへの追加 [17](#)

## こ

コマンドタスク

エラー [126](#)

コマンドの構文 [62](#)

シェルコマンド [62](#)

詳細プロパティ [65](#)

設定プロパティ [65](#)

トラブルシューティング [65](#)

入力 [63](#)

ワークフローパラメータ [63](#)

ワークフロー変数 [63](#), [64](#)

概要 [62](#)

出力 [64](#)

## し

シーケンスフロー

概要 [14](#)

ゲートウェイ [123](#)

条件 [15](#)

条件付きシーケンスフロー [123](#)

シェルコマンド

ワークフローでの実行 [62](#)

式

条件付きシーケンスフロー [15](#)

割り当てタスク [30](#)

失敗したタスク  
条件付きシーケンスフロー [15](#)  
自動ワークフローリカバリ  
およびヒューマンタスク [127](#)  
自動割り当て  
説明 [89](#)  
終結イベント [13](#)  
終了イベント [13](#)  
受信者  
通知タスク [112](#)  
動的な電子メール [113](#)  
出力式  
設定方法 [98](#)  
条件  
シーケンスフロー [15](#)  
条件付きシーケンスフロー  
概要 [15](#)  
失敗したタスク [15](#)  
タスク出力 [15](#)  
ワークフローパラメータ [16](#)  
ワークフロー変数 [16](#)

## す

スキップリカバリストラテジ  
タスク [129](#)  
すべてクリア  
マッピング入力 [89](#)

## せ

成功  
出力 [15](#)  
[全般] ビュー  
マッピングタスク [85](#)

## た

タスク  
アクティビティ [13](#)  
エラー [126](#)  
概要 [13](#)  
監視 [24](#)  
クラスタの作成 [52](#)  
クラスタの削除 [52](#)  
コマンド [62](#)  
出力 [28](#)  
状態 [145](#)  
通知 [111](#)  
ヒューマン [66](#)  
マッピング [84](#)  
ログ [24](#)  
ワークフローへの追加 [17](#)  
割り当て [30](#)  
入力 [28](#)  
タスク出力  
概要 [28](#)  
条件付きシーケンスフロー [15](#)  
成功 [15](#)  
ワークフロー変数 [28, 31](#)  
ワークフロー変数での取得 [32](#)  
タスクの実行者  
ヒューマンタスクのロール [69, 75](#)  
タスクリカバリストラテジ  
概要 [129](#)

タスクリカバリストラテジ (続く)  
設定 [133](#)  
マッピングタスク [129](#)  
タスク入力  
ワークフローパラメータ [39](#)  
ワークフロー変数 [28](#)  
概要 [28](#)

## つ

通知タスク  
エラー [126](#)  
概要 [111](#)  
受信者 [112](#)  
詳細プロパティ [118](#)  
電子メールアドレス [114](#)  
トラブルシューティング [118](#)  
ワークフローパラメータ [111](#)  
ワークフロー変数 [111, 117](#)  
出力 [117](#)  
電子メールのコンテンツ [116](#)

## て

電子メール  
アドレス [114](#)  
受信者 [112](#)  
動的な受信者 [113](#)  
動的な電子メールアドレス [115](#)  
動的なコンテンツ [116](#)

## と

動的な受信者  
電子メール [113](#)  
動的な電子メール  
アドレス [115](#)  
コンテンツ [116](#)  
受信者 [113](#)

## に

入力  
コマンドタスク [63](#)  
ヒューマンタスク [73](#)

## は

バインディング  
出力のワークフロー変数へのマッピング [102](#)  
説明 [91](#)  
パラメータ  
ワークフロー [38](#)  
パラメータの使用  
マッピングタスクの入力での表示 [88](#)  
パラメータファイル  
Developer tool からのエクスポート [50](#)  
作成 [50](#)  
サンプル [49](#)  
ヒント [49](#)  
マッピングの実行 [51](#)  
ルール [49](#)  
ワークフロー [38](#)

パラメータファイル (続く)  
ワークフローの実行時に使用 [48](#)  
目的 [48](#)  
パラメータセット  
作成 [46](#)  
マッピングタスク [85](#)  
ワークフローへの割り当て [38](#)  
概要 [44](#)

## ひ

ビジネス管理者  
ヒューマンタスクのロール [69](#), [75](#)  
ヒューマンタスク  
エラー [126](#)  
概要 [66](#)  
出力 [73](#)  
ステップのプロパティ [74](#)  
タスクインスタンスの設定 [79](#)  
タスクステップの設定 [80](#)  
タスクの実行者 [69](#), [75](#)  
[タスク分散] タブ [72](#)  
入力 [73](#)  
ビジネス管理者 [69](#), [75](#)  
ヒューマンタスクのプロパティ [69](#)  
ワークフロー変数 [73](#)

## へ

変数  
ワークフロー [27](#)

## ほ

方法  
保持された出力を入力タスクにバインドする [101](#)  
マッピング出力の設定 [95](#)  
マッピング出力の保持 [100](#)  
マッピング出力のワークフロー変数へのバインド [102](#)  
マッピングでの出力式の定義 [98](#)

## ま

マッピングタスク  
エラー [126](#)  
概要 [84](#)  
クラスタワークフロー内のマッピングタスク [110](#)  
[全般] ビュー [85](#)  
入力 [86](#)  
バインド出力 [93](#)  
パラメータセット [85](#)  
マッピングパラメータのオーバーライド [87](#)  
ログファイル [103](#)  
ワークフローに複数含める [84](#)  
ワークフロー変数 [90](#)  
出力 [90](#)  
詳細プロパティ [104](#)  
設定プロパティ [104](#)  
マッピングタスクの入力  
保持された出力をバインド [101](#)  
マッピング入力  
自動割り当てオプション [89](#)  
割り当てのクリア [89](#)

マッピングパラメータ  
パラメータの使用 [88](#)  
ワークフローでのオーバーライド [87](#)  
ワークフローでの使用 [87](#)  
マッピング出力  
infacmd の使用 [92](#)  
出力の定義方法 [97](#)  
設定手順 [95](#)  
デプロイメントの変更 [93](#)  
保持のガイドライン [93](#)  
ワークフロー変数にバインド [93](#)  
保持 [91](#)  
マッピング出力の保持  
infacmd の使用 [92](#)  
タスク入力へのバインド [101](#)  
方法 [100](#)  
ルールおよびガイドライン [93](#)  
説明 [91](#)

## り

リカバリ  
タスク [129](#)  
ヒューマンタスク [135](#)  
マッピングタスク [135](#)  
ワークフロー [125](#), [141](#)

## ろ

ログ  
ワークフロー [148](#)  
ワークフローインスタンス [24](#)  
ログイベント  
ワークフロー [148](#)  
ログファイル  
マッピングタスク [103](#)

## わ

ワークフロー  
アクティビティ [13](#)  
イベント [13](#)  
インスタンス [23](#)  
エラー [126](#)  
オブジェクトの追加 [17](#)  
概要 [11](#)  
監視 [24](#)  
キャンセル [146](#)  
強制終了 [146](#)  
クラスタ [52](#), [61](#)  
ゲートウェイ [14](#)  
検証 [18](#)  
作成 [16](#)  
シーケンスフロー [14](#)  
シェルコマンドの実行 [62](#)  
実行 [23](#)  
条件付きシーケンスフロー [15](#)  
詳細プロパティ [21](#)  
タスク [13](#)  
デプロイ [23](#)  
トレースレベル [21](#)  
パラメータ [38](#)  
パラメータファイル [38](#)  
リカバリ [148](#)  
リカバリプロパティ [21](#)



#### ワークフロー (続く)

ログ [24](#), [148](#)

削除 [24](#)

状態 [144](#)

#### ワークフローインスタンス

監視 [24](#)

実行 [23](#)

定義 [23](#)

リカバリ [137](#)

ログ [24](#)

#### ワークフローの監視

概要 [141](#)

#### ワークフローの状態

リカバリ可能 [126](#)

#### ワークフローパラメータ

エスケープ文字 [41](#)

コマンドタスク [63](#)

作成 [42](#)

タスク入力 [39](#)

通知タスク [111](#)

データ型の変換 [44](#)

デフォルト値 [43](#)

電子メールアドレス [115](#)

電子メールの受信者 [113](#)

ネストされた [42](#)

命名規則 [41](#)

リカバリ [134](#)

割り当てタスク [31](#)

条件付きシーケンスフロー [16](#)

電子メールのコンテンツ [116](#)

#### ワークフローリカバリ

概要 [125](#), [147](#)

グリッド [127](#)

実行 [148](#)

自動 [127](#)

詳細プロパティ [21](#)

設定 [133](#)

タスクエラー [126](#)

タスクリカバリストラテジ [129](#)

#### ワークフローリカバリ (続く)

リカバリ可能な状態 [126](#)

ログ [135](#)

ワークフローエラー [126](#)

ワークフローパラメータ [134](#)

ワークフロー変数 [134](#)

#### ワークフロー変数

エスケープ文字 [34](#)

コマンドタスク [63](#), [64](#)

システム [29](#)

初期値 [29](#)

説明 [27](#)

タイプ [27](#)

タスク出力 [28](#), [31](#)

タスク出力の取得 [32](#)

タスク入力 [28](#)

通知タスク [111](#), [117](#)

データ型変換 [36](#)

電子メールアドレス [115](#)

電子メールの受信者 [113](#)

ネストされた [35](#)

ヒューマンタスク [73](#)

へのマッピング出力のバインディング [102](#)

マッピングタスクの出力 [90](#)

命名規則 [34](#)

ユーザー定義変数の作成 [29](#)

ユーザー定義 [29](#)

リカバリ [134](#)

割り当てタスク [30](#), [31](#)

条件付きシーケンスフロー [16](#)

値の変更 [31](#)

電子メールのコンテンツ [116](#)

日付形式 [36](#)

#### 割り当てタスク

概要 [30](#)

ワークフローパラメータ [31](#)

ワークフロー変数 [30](#), [31](#)

構成 [31](#)