



Informatica® Cloud Application Integration  
October 2018

**チュートリアル: 電卓**

Informatica Cloud Application Integration チュートリアル: 電卓  
October 2018  
2018 年 11 月

© 著作権 Informatica LLC 1993, 2019

発行日: 2019-02-11

# 目次

<b>序文</b> .....	4
<b>第 1 章：概要</b> .....	5
作業を開始する前に.....	6
概念、アセット、およびサービス.....	6
チュートリアル目的.....	7
<b>第 2 章：サービスコネクタと接続の作成</b> .....	8
手順 1: サービスコネクタを作成するための WSDL ファイルのインポート.....	8
サービスコネクタのプロセスオブジェクト.....	10
手順 2: アクションのテスト.....	11
手順 3: 接続の作成.....	12
<b>第 3 章：プロセスの作成</b> .....	15
手順 1: プロセスの作成.....	15
手順 2: 入力の設定.....	17
手順 3: 並列で実行されるサービスの設定.....	18
手順 4: 出力の設定.....	20
<b>第 4 章：プロセスの起動</b> .....	22
手順 1: 整数 10 と 5 を使用したプロセスの起動.....	22
プロセスインスタンスのアプリケーション統合コンソールでの表示.....	23
手順 2: 整数 8 と 0 を使用したプロセスの起動.....	23
フォールトになったプロセスインスタンスのアプリケーション統合コンソールでの表示.....	24
<b>第 5 章：フォールト処理を有効化するためのプロセスの更新</b> .....	25
手順 1: フォールト処理の有効化.....	25
手順 2: 出力設定の変更.....	27
<b>第 6 章：整数 8 と 0 を使用した更新後のプロセスの起動</b> .....	30
更新されたプロセスインスタンスのアプリケーション統合コンソールでの表示.....	30
<b>第 7 章：オプション：結果のアーカイブ</b> .....	32
JDBC 接続の作成.....	32
結果のアーカイブ.....	34

# 序文

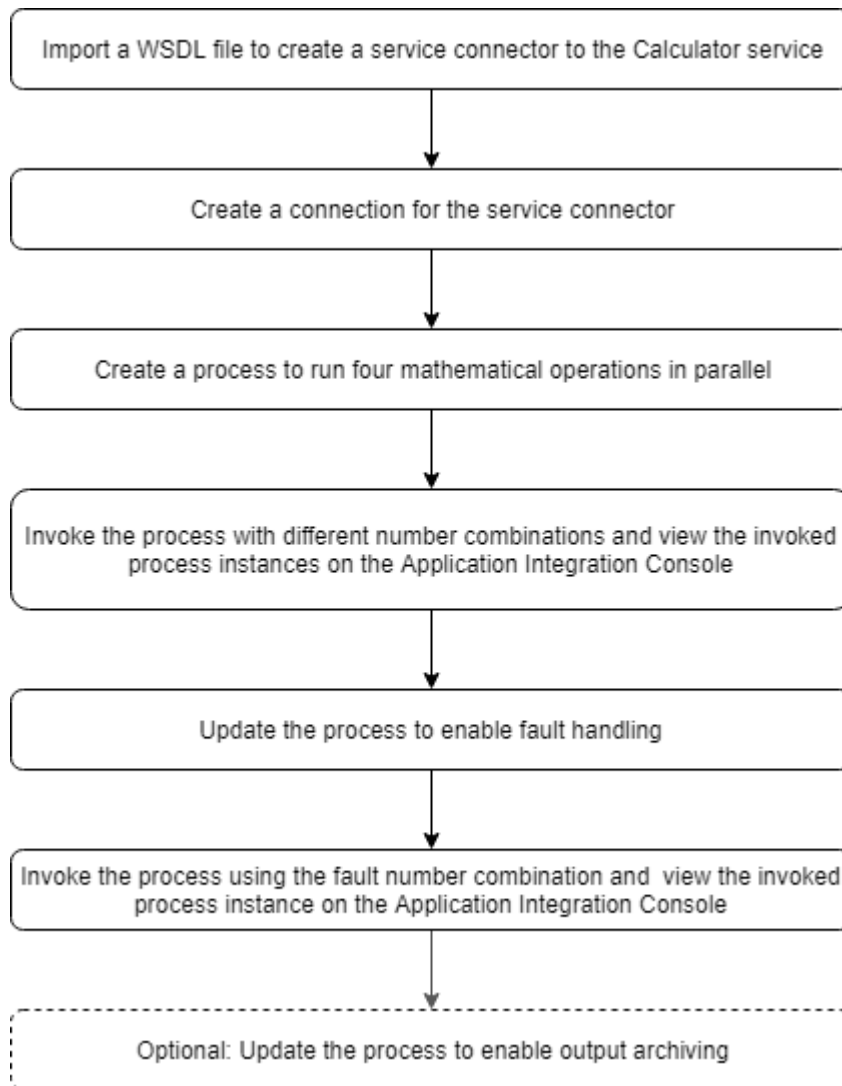
このモジュールは、4つの演算を並行して実行するプロセスの作成および起動に関するチュートリアルです。

# 第 1 章

## 概要

電卓のチュートリアルでは、アプリケーションの統合を使用して、4つの算術演算を並行して実行するプロセスを作成します。異なる整数を使用してプロセスを起動し、エラーが発生した場合のフォールト処理を有効にします。また、アプリケーション統合コンソールで、起動されたプロセスインスタンスを表示できます。

次の図は、電卓のチュートリアルで行うプロセスを順を追って示しています。



# 作業を開始する前に

開始する前に、次のタスクを実行します。

1. Informatica Intelligent Cloud Services アカウントを作成して、アプリケーションの統合サービスとアプリケーション統合コンソールサービスにアクセスできることを確認します。アカウント作成方法の詳細については、[Training Program Overview](#) コミュニティの記事の、登録に関するセクションを参照してください。
2. アプリケーションの統合にログインし、**[参照]** ページに移動して、参照リストから **[プロジェクト]** を選択します。
3. チュートリアルプロジェクトを作成します。**[新規プロジェクト]** をクリックして、プロジェクトを作成します。
4. **[チュートリアル]** 内に **[電卓]** フォルダを作成します。**[新規フォルダ]** をクリックして、フォルダを作成します。

## 概念、アセット、およびサービス

このチュートリアルで作成するアセットと、使用するサービスと概念について説明します。

### アセット

#### サービスコネクタ

サービスコネクタは、Informatica Intelligent Cloud Services のサービスに公開済み API がある場合、そのサービスと、ファイアウォール内またはクラウドの任意の場所で連携します。このチュートリアルでは、サービスコネクタを新規では作成しません。代わりに、WSDL ファイルをインポートしてサービスコネクタを作成します。

#### サービスコネクタのプロセスオブジェクト

プロセスオブジェクトは、データをグループ化および構造化します。サービスコネクタのプロセスオブジェクトは、サービスコネクタの入力と出力をグループ化および構造化します。サービスコネクタのプロセスオブジェクトは、サービスコネクタを使用するすべてのプロセスで使用できます。例えば、名前、住所、電話番号などの人口統計情報をサービスが返す場合、この出力情報を保存するために、「人口統計」という名前のプロセスオブジェクトを作成できます。また、サービスコネクタとは関係ないプロセスオブジェクトを作成することもできます。このチュートリアルでは、サービスコネクタ内に作成するプロセスオブジェクトを扱います。

#### 接続

接続は、サービスコネクタに接続するための設定を定義します。サービスコネクタは、JDBC などのコネクタに構築する場合と、このチュートリアルの電卓サービスコネクタのように、自作のコネクタに構築する場合があります。プロセスでサービスコネクタを使用するには、まず接続を作成し、次にプロセスに接続を追加します。

#### プロセス

プロセスは、ビジネス要件を自動化する命令のセットです。プロセスステップを使用して、タスクを順番に実行し、複雑な XQuery を使用して情報を取得し、いくつかの基準に基づいた決断などを行うプロセスを設定します。

## 概念

### フォールト処理

フォールト処理は、プロセスフローの間に発生する例外や予期しない条件を解決するようにプロセスを設定することを指します。フォールト処理を行うことで、常に予測可能な結果が発生します。例えば、出力変数にエラーが発生する可能性がある場合、フォールト処理を有効化すると「HTTPS 500 エラー」の代わりに「フォールト」というテキストが表示されます。

## サービス

### アプリケーションの統合

アプリケーションの統合は、API ベースの統合、イベント処理、サービスとプロセスのオーケストレーションを実行するために使用できる、Informatica Intelligent Cloud Services のサービスです。

### アプリケーション統合コンソール

アプリケーション統合コンソールは、起動されたプロセスインスタンスの詳細な監視やデバッグを実行するために使用できる、Informatica Intelligent Cloud Services のサービスです。

# チュートリアルの目的

このチュートリアルを終了すると、次のタスクを実行できるようになります。

- WSDL ファイルをインポートして、サービスコネクタを作成する。
- サービスコネクタ用の接続を作成する。
- 次のステップを使用するプロセスを作成する。
  - 割り当て
  - 作成
  - データディシジョン
  - 並列パス
  - サービス
- フォールト処理を有効にするために、プロセスを更新する。
- 起動したプロセスインスタンスをアプリケーション統合コンソールに表示する。
- 作成ステップを使用してプロセス出力をアーカイブする。このタスクはオプションです。

## 第 2 章

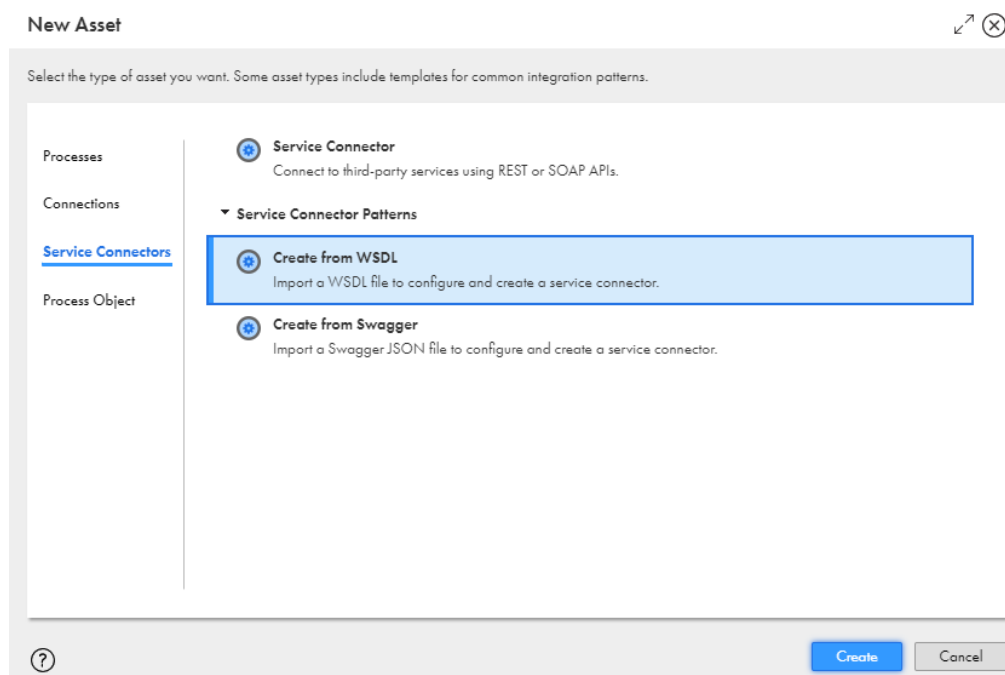
# サービスコネクタと接続の作成

<http://www.dneonline.com/calculator.asmx> の電卓 Web サービスが提供する加算、減算、乗算、除算のサービスにアクセスするための、サービスコネクタを作成します。WSDL ファイルをインポートして、サービスコネクタを作成します。WSDL ファイルをインポートしたら、サービスコネクタを完成させるために、簡単なタスクを手動で実行する必要があります。



サービスコネクタの作成後は、プロセスでサービスコネクタを使用できるようにするために、接続を設定する必要があります。

## 手順 1: サービスコネクタを作成するための WSDL ファイルのインポート

1. アプリケーションの統合にログインして、[新規] をクリックします。
2. [新しいアセット] ダイアログボックスで、[サービスコネクタ] > [サービスコネクタパターン] > [WSDL からの作成] の順に選択し、[作成] をクリックします。



3. **【WSDL からの新規コネクタ】** ダイアログボックスの **【WSDL ソース】** タブに次のプロパティを入力します。
- a. **【名前】** フィールドに、「**CalculatorSVC**」と入力します。
  - b. **【プロジェクト】** フィールドの横で、**【チュートリアル】** > **【電卓】** の順にフォルダを移動して選択します。
  - c. **【説明】** フィールドに、「**dneonline.com/calculator.asmx の電卓 Web サービスからアクションを起動**」とテキストを入力します。
  - d. **【WSDL ソース】** の横で、**【URL】** を選択し、次の URL を **【URL】** フィールドに貼り付けます：  
<http://www.dneonline.com/calculator.asmx?WSDL>。
  - e. このサービスには認証が必要ないため、**【認証の使用】** オプションを無効化します。

**New Service Connector from WSDL**  

1 **WSDL Source** 2 **Service and Operations** 3 **Summary**

Provide identifying information for the service connector and WSDL file.

Name: \*

Location: \*

Description:

WSDL File

WSDL source: \* ☐ File ☒ URL

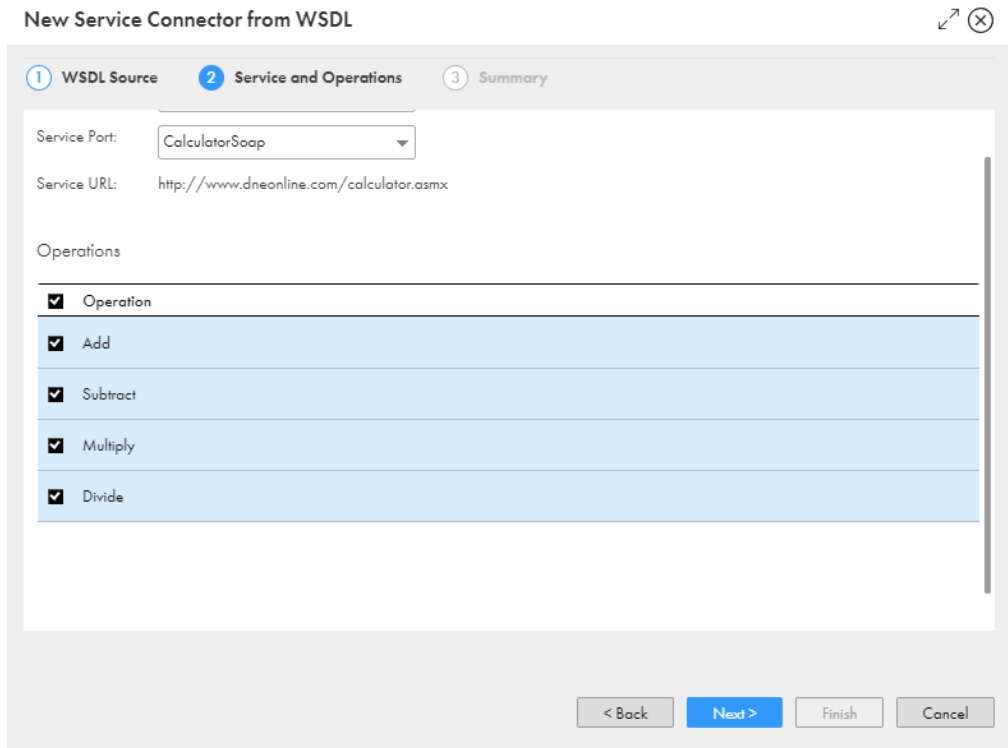
☐ Use authentication

User Name:

Password:

4. **【次へ】** をクリックします。

**【サービスと操作】** タブが表示されます。電卓サービスは、**加算、減算、乗算、および除算**の演算を実行するために使用できます。このタブの値は何も変更しないようにします。



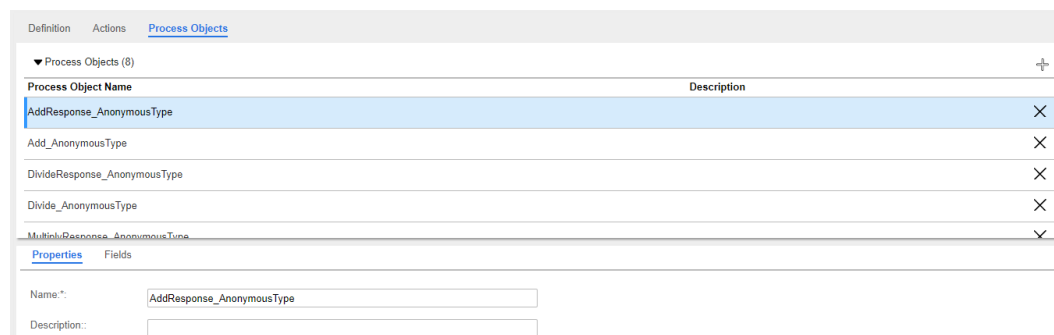
5. [次へ] をクリックします。  
[サマリ] タブには、サービスコネクタの詳細と、次に行う必要がある指示が表示されます。
6. [完了] をクリックします。  
生成したサービスコネクタが表示された状態で Process Designer が表示されます。

## サービスコネクタのプロセスオブジェクト

サービスコネクタのプロセスオブジェクトは、データをグループ化します。サービスコネクタを生成するためにインポートした WSDL ファイルは、8 つのプロセスオブジェクトを生成します。

サービスコネクタのプロセスオブジェクトを表示するには、**電卓サービス**の【プロセスオブジェクト】タブをクリックします。

次の図は、**電卓サービス**コネクタの【プロセスオブジェクト】タブを示します。



次の表では、サービスコネクタのプロセスオブジェクトについて説明しています。

名前	説明	フィールド数	フィールド名	フィールドの型
AddResponse_AnonymousType	加算アクションの結果が格納されます。	1 つ	AddResult	number
Add_AnonymousType	加算アクションの実行に必要な入力格納されます。	2 つ	intA、intB	number
DivideResponse_AnonymousType	除算アクションの結果が格納されます。	1 つ	DivideResult	number
Divide_AnonymousType	除算アクションの実行に必要な入力格納されます。	2 つ	intA、intB	number
MultiplyResponse_AnonymousType	乗算アクションの結果が格納されます。	1 つ	MultiplyResult,	number
Multiply_AnonymousType	乗算アクションの実行に必要な入力格納されます。	2 つ	intA、intB	number
SubtractResponse_AnonymousType	減算アクションの結果が格納されます。	1 つ	SubtractResult	number
Subtract_AnonymousType	減算アクションの実行に必要な入力格納されます。	2 つ	intA、intB	number

**注:** Add\_AnonymousType、Divide\_AnonymousType、Multiply\_AnonymousType、および Subtract\_AnonymousType の各プロセスオブジェクトに含まれるフィールドは同一です。これは、加算、除算、乗算、および減算の各アクションの入力に、同じ数を使用しているためです。

## 手順 2: アクションのテスト

インポートした WSDL ファイルには、サービスコネクタの生成に必要な情報のほとんどが含まれています。テスト値を追加して、各アクションをテストする必要があります。

1. **【アクション】** タブに移動して、**【加算】** アクションを選択します。
2. **【プリエンティブ認証】** オプションをクリアします。
3. **【入力】** タブを選択し、**【テストに使用する値】** フィールドに次の値を入力します: { "intA": 10, "intB": 2 }。
4. **【テスト】** タブを選択して、次に **【テスト】** をクリックします。  
テストが成功し、出力 12 が返されます。

Action Input Binding Output **Test**

---

Test Server:\* Test on Cloud Server ▼ Test

Result: ✓ Successful.

HTTP Status: 200

URL: http://www.dneonline.com/calculator.asmx

**Output Fields**

Output	Value
AddResponse	<AddResponse><AddResult>12</AddResult></AddResponse>

▼ Response Payload:

```

1 <soap:Envelope xmlns:rest="http://schemas.activebpel.org/REST/2007/12/01/aeREST.xsd"
2   xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"

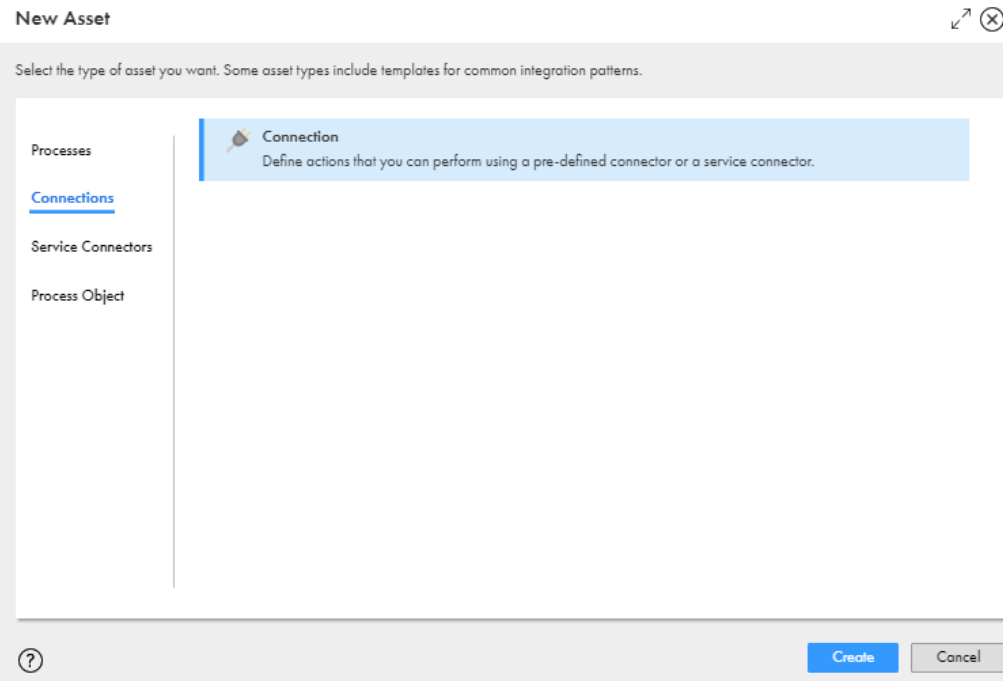
```

- 減算、乗算、および除算のそれぞれのアクションについて、[1](#) から [4](#) までの手順を繰り返します。
  - 【アクション】 > 【テスト】 タブに移動して、【テスト】 をクリックします。  
【成功】の結果が表示されます。
  - 【アクション】 > 【パブリッシュ】 をクリックします。
- 電卓サービスにアクセスするためのサービスコネクタが作成されました。

## 手順 3: 接続の作成


プロセスでサービスコネクタを使用する前に、サービスコネクタの接続を作成します。接続プロパティを入力し、接続を保存して、接続をパブリッシュします。

- アプリケーションの統合で、【新規】 をクリックします。
- 【新しいアセット】 ダイアログボックスで、【接続】 > 【接続】 の順に選択し、【作成】 をクリックします。



接続テンプレートが表示された、Process Designer が開きます。

3. 【プロパティ】 タブに次の情報を入力します。
  - a. 【名前】 フィールドに、「**CalculatorConn**」と入力します。
  - b. 【プロジェクト】 フィールドの横で、【参照】 をクリックしたら、【チュートリアル】 > 【電卓】 の順にフォルダを移動して選択します。
  - c. 【説明】 フィールドに、「**電卓サービスが提供する加算、減算、乗算、および除算のアクションを使用**」というテキストを入力します。
  - d. 【タイプ】 リストで、【チュートリアル】 > 【電卓】 > 【CalculatorSVC】 の順に選択します。
  - e. 【実行】 リストで、【Cloud Server または任意の Secure Agent】 を選択します。
4. 【保存】 をクリックして接続を保存します。
5. 【アクション】 > 【パブリッシュ】 をクリックします。

CalculatorConn |  Valid

[Properties](#) Metadata

---

**Connection Details**

---

Name:*	CalculatorConn	(Unpublish connection to edit name)
Location:	Tutorials\Calculator	<a href="#">Browse...</a>
Description:	Use the Add, Subtract, Multiply, and Divide Actions provided by Calculator Service.	
Type:*	Tutorials > Calculator > CalculatorSVC ▼	
Run On:	Cloud Server or any Secure Agent ▼	
Connection Test:	Not Supported	
OData-Enabled:	Not Supported	

## 第 3 章

# プロセスの作成

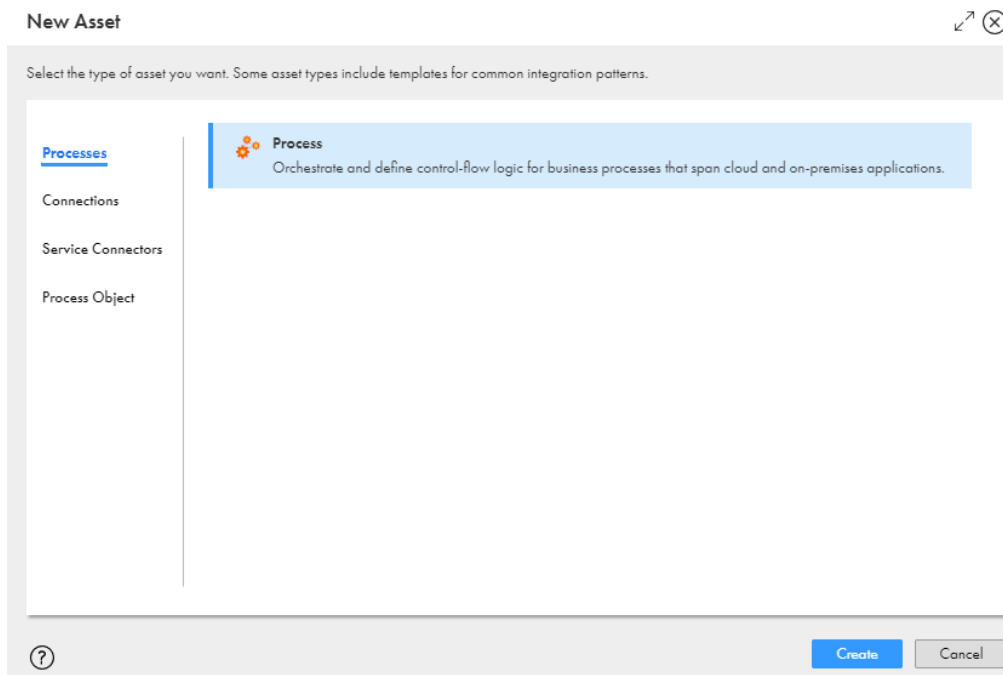
電卓サービスが提供する加算、除算、乗算、および減算の 4 つのアクションを実行するプロセスを作成します。このプロセスは、入力として 2 つの数値を取り、この 2 つの数値に対して各演算を並行して実行します。

## 手順 1: プロセスの作成

プロセスを作成して【開始】ステップを使用して、プロセスの基本プロパティを設定します。

1. アプリケーションの統合で、【新規】をクリックします。
2. 【新しいアセット】ダイアログボックスで、【プロセス】 > 【プロセス】の順に選択し、【作成】をクリックします。

プロセステンプレートが表示された、Process Designer が開きます。



3. 【開始】ステップをクリックします。  
プロセスのプロパティセクションが表示されます。

4. **【全般】** タブを選択し、次のプロパティを入力します。
  - a. **【名前】** フィールドに、「**電卓**」と入力します。
  - b. **【プロジェクト】** の横にある **【選択】** をクリックし、**【チュートリアル】** > **【電卓】** の順にフォルダを移動して選択します。
  - c. **【説明】** フィールドに、「**電卓サービスを使用して、4つの算術演算を並列に実行**」と入力します。
5. **【開始】** タブを選択し、次のプロパティを入力します。
  - a. **【バインディング】** リストで、**【HTTP/SOAP】** を選択します。  
バインディングプロパティは、プロセスの起動方法を定義します。
  - b. **【許可されたロール】** の横で **【サービスコンシューマ】** と指定します。  
サービスコンシューマのロールが割り当てられているユーザーのみが、プロセスサービスの URL にアクセスし、プロセスを起動できます。
  - c. **【実行】** リストで、**【Cloud Server】** を選択します。  
**【開始】** タブの **【適用対象】** フィールドまたは **【実行】** フィールドは変更しないようにします。
6. **【入力フィールド】** タブを選択し、次のフィールドを追加してそれぞれのタイプを設定します。

名前	タイプ
NumberA	Integer
NumberB	Integer

7. **【出力フィールド】** タブを選択し、次のフィールドを追加してそれぞれのタイプを設定します。

名前	タイプ
和	Integer
差	Integer
積	Integer
商	Integer

これらのフィールドは、プロセスで設定した各算術演算の出力を表示します。

8. **【一時フィールド】** タブを選択し、次のフィールドを追加してそれぞれのタイプを設定します。

名前	タイプ
add_input	タイプをさらに表示 > 定義された接続タイプ > CalculatorSVC > CalculatorConn > Add_AnonymousType
subtract_input	タイプをさらに表示 > 定義された接続タイプ > CalculatorSVC > CalculatorConn > Subtract_AnonymousType

名前	タイプ
multiply_input	タイプをさらに表示 > 定義された接続タイプ > CalculatorSVC > CalculatorConn > Multiply_AnonymousType
divide_input	タイプをさらに表示 > 定義された接続タイプ > CalculatorSVC > CalculatorConn > Divide_AnonymousType

サービスコネクタが想定する入力の形式は、2つの数値を格納するプロセスオブジェクトです。各アクションに必要な入力を持つための一時フィールドを定義します。

9. **【詳細】** タブを選択し、**【トレースレベル】** を **【詳細】** に設定します。

トレースレベルが詳細に設定されている場合、アプリケーション統合コンソールはすべてのステップを記録します。これは、プロセスのデバッグを行う必要がある場合に便利です。

## 手順 2: 入力の設定

割り当てステップを使用して、入力フィールドを定義します。入力値は、プロセスの起動時のランタイムに入力します。

1. **【割り当て】** ステップをクリックし、パレットから **【開始】** ステップと **【終了】** ステップの間のキャンバス上にドラッグします。
2. **【割り当て】** ステップを選択します。  
ステップのプロパティセクションが表示されます。
3. **【全般】** タブをクリックし、**【名前】** フィールドに「**入力の設定**」と入力します。
4. **【割り当て】** タブをクリックし、次のフィールドを追加してそのタイプと値を設定します。


フィールド	タイプ	値
add_input > intA	フィールド	NumberA
add_input > intB	フィールド	NumberB
subtract_input > intA	フィールド	NumberA
subtract_input > intB	フィールド	NumberB
multiply_input > intA	フィールド	NumberA
multiply_input > intB	フィールド	NumberB
divide_input > intA	フィールド	NumberA
divide_input > intB	フィールド	NumberB









**注:** 入力フィールドは、手順 1 のプロセスの基本プロパティの設定で定義したため、選択のために使用できるようになっています。

プロセスを起動する際に **NumberA** と **NumberB** の値を指定します。

5. **【保存】** をクリックします。

次の図は、**【入力の設定】** ステップのプロパティセクションを示します。

 Configure Input Properties

General	Target	Value	
<b>Assignments</b>	add_input > intA	Field ▼	NumberA ▼ 
	add_input > intB	Field ▼	NumberB ▼ 
	subtract_input > intA	Field ▼	NumberA ▼ 
	subtract_input > intB	Field ▼	NumberB ▼ 
	multiply_input > intA	Field ▼	NumberA ▼ 
	multiply_input > intB	Field ▼	NumberB ▼ 
	divide_input > intA	Field ▼	NumberA ▼ 
	divide_input > intB	Field ▼	NumberB ▼ 

**Add Field ▼** **Reorder**

プロセスの入力の定義が完了しました。次に、並行して実行するサービスのオーケストレーションを行います。

## 手順 3: 並列で実行されるサービスの設定

並列パスとサービスのステップを使用して、加算、減算、乗算、および除算の演算を同時に実行します。

1. **【並列パス】** ステップを、パレットから **【開始】** ステップと **【終了】** ステップの間のキャンバス上にドラッグします。
2. **【並列パス】** ステップを選択します。  
ステップのプロパティセクションが表示されます。
3. **【全般】** タブをクリックし、**【名前】** フィールドに「**並列演算**」と入力します。
4. **【並列パス】** タブをクリックして、次のタスクを実行します。
  - a. 4 つのパスを追加します。
  - b. 各パスに**サービスステップ**をドラッグします。**【サービスタイプ】** を **【接続】** に設定し、**【接続】** を **【チュートリアル】** > **【CalculatorSVC】** > **【CalculatorConn】** に設定します。
  - c. 最初のパスに**加算**アクションを選択し、入力フィールド **【加算】** を追加して、タイプを **【フィールド】** に、値を **add\_input** に設定します。
  - d. 残り 3 つのパスに手順 4c を繰り返します。

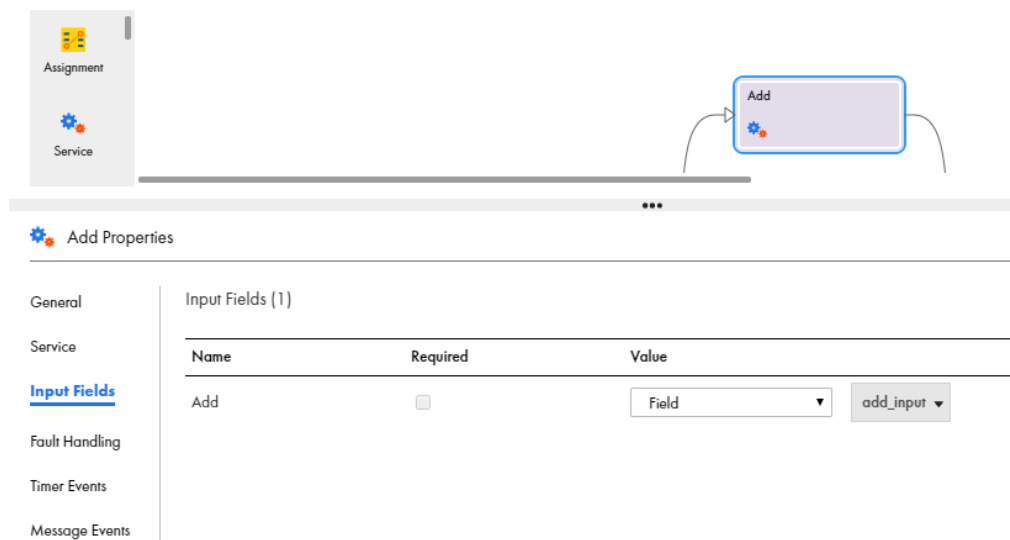
次の表は、各パスのサービスステップに追加する必要があるアクションを示します。

パス	サービスタイプ	接続	アクション	入力フィールド
パス 0	接続	チュートリアル > CalculatorSVC > CalculatorConn	加算	加算 > フィールド > add_input
パス 1	接続	チュートリアル > CalculatorSVC > CalculatorConn	減算	減算 > フィールド > subtract_input
パス 2	接続	チュートリアル > CalculatorSVC > CalculatorConn	乗算	乗算 > フィールド > multiply_input
パス 3	接続	チュートリアル > CalculatorSVC > CalculatorConn	除算	除算 > フィールド > divide_input

次の図は、**【並列パス】** ステップのステッププロパティのセクションを示します。

Parallel Operations Properties	
General	Paths
<b>Parallel Paths</b>	
Path 0	Path starting with Service ( Add )
Path 1	Path starting with Service ( Subtract )
Path 2	Path starting with Service ( Multiply )
Path 3	Path starting with Service ( Divide )

次の図は、**加算** サービスステップのプロパティセクションを示します。



5. **【保存】** をクリックします。

## 手順 4: 出力の設定

割り当てステップを使用して、プロセスを起動する際に表示する出力を定義します。加算、減算、乗算、および除算のサービスごとに出力フィールドが必要です。

1. **【割り当て】** ステップをクリックし、パレットから **【並列パス】** ステップと **【終了】** ステップの間のキャンバス上にドラッグします。
2. **【割り当て】** ステップを選択します。  
ステップのプロパティセクションが表示されます。
3. **【全般】** タブをクリックし、**【名前】** フィールドに「**出力の設定**」と入力します。
4. **【割り当て】** タブをクリックして、以下のフィールドを追加します。

フィールド	タイプ	値
和	フィールド	AddResponse > AddResult
差	フィールド	SubtractResponse > SubtractResult
積	フィールド	MultiplyResponse > MultiplyResult
商	フィールド	DivideResponse > DivideResult

5. **【保存】** をクリックします。
  6. **【検証】** ボタンをクリックします。エラーがない場合には、**【保存】** をクリックします。  
エラーがある場合は、修正してプロセスを保存します。
  7. **【アクション】** > **【パブリッシュ】** をクリックします。
- 次の図は、**【出力の設定】** ステップのプロパティセクションを示します。

## Configure Output Properties

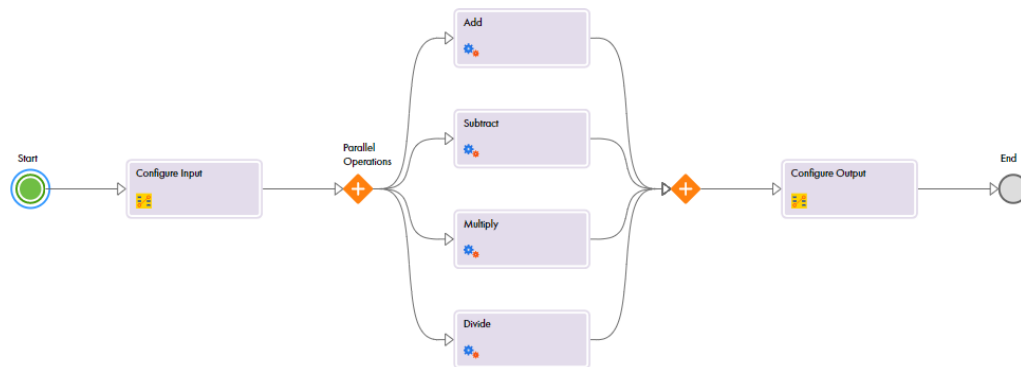
General

Assignments

Target	Value	
Sum	Field	AddResponse > AddResult
Difference	Field	SubtractResponse > SubtractResult
Product	Field	MultiplyResponse > MultiplyResult
Quotient	Field	DivideResponse > DivideResult

Add Field
Reorder

次の図は、完成したプロセスを示しています。



## 第 4 章

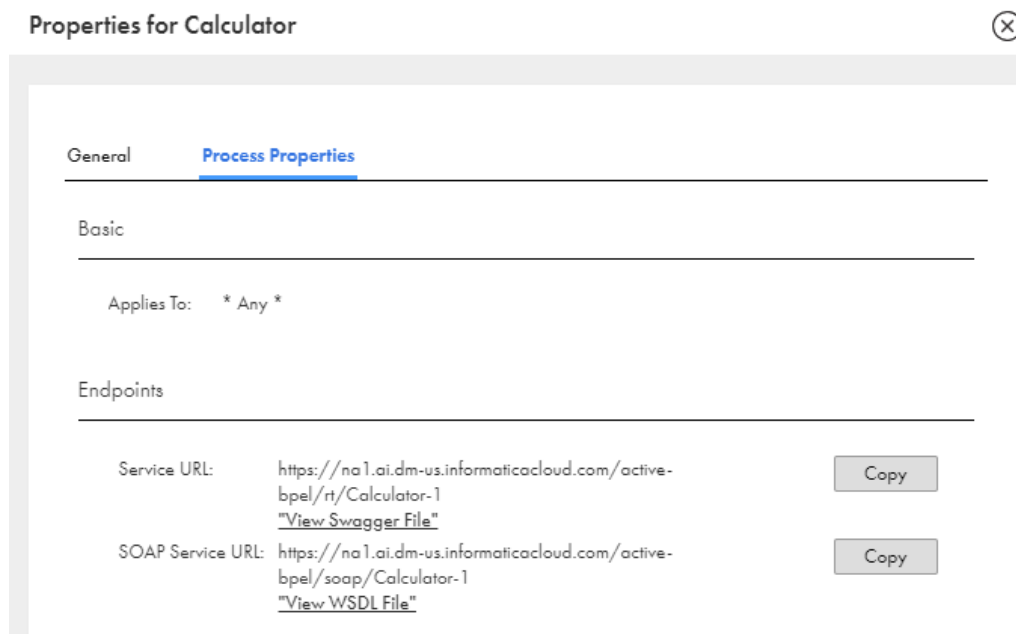
# プロセスの起動

2つの整数のセット（10 と 5、8 と 0）を使用して、プロセスを起動します。アプリケーション統合コンソールで、取得した出力とプロセスインスタンスを表示します。

## 手順 1: 整数 10 と 5 を使用したプロセスの起動

整数 10 と 5 を使用して、プロセスサービスの URL をブラウザで起動します。プロセスは正常に起動されます。

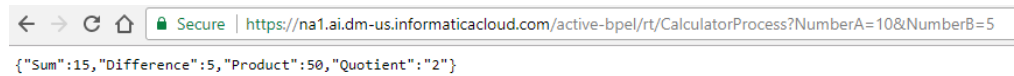
1. アプリケーションの統合で、プロセスの横の【アクション】 > 【プロパティ】をクリックします。  
これは、Process Designer または【参照】ページから行えます。  
【電卓のプロパティ】ウィンドウが表示されます。
2. 【プロセスのプロパティ】タブを選択し、サービス URL をコピーします。



3. ブラウザを開き、検索バーに次のリンクを貼り付けて **Enter** を押します。  
<Service URL>?NumberA=10&NumberB=5  
例えば、次の URL を貼り付けます。  
<https://na1.ai.dm-us.informaticacloud.com/active-bpel/rt/CalculatorProcess-1?NumberA=10&NumberB=5>

次の値で応答が表示されます。

- 和: 15
- 差: 5
- 積: 50
- 商: 2



## プロセスインスタンスのアプリケーション統合コンソールでの表示

- ▶ **【マイサービス】** リストをクリックして、アプリケーション統合コンソールを選択します。

アプリケーション統合コンソールが**【プロセス】** ページに表示されます。電卓プロセスの**【状態】** が**【完了】** になっています。

The screenshot shows the 'Processes' page in the Application Integration Console. It displays a table with one process instance. The table has columns for Id, Name, Version, Start Date, End Date, and State. The process instance is 'Calculator' with version '13', started on '2018/4/25 11:14:17:12 AM', and ended on '2018/4/25 11:14:19:19 AM' with a state of 'Completed'.

Id	Name	Version	Start Date	End Date	State
173753580224794624	Calculator	13	2018/4/25 11:14:17:12 AM	2018/4/25 11:14:19:19 AM	Completed

## 手順 2: 整数 8 と 0 を使用したプロセスの起動

整数 8 と 0 を使用して、プロセスサービスの URL を起動します。0 による除算はできないため、エラーが表示されます。

1. ブラウザを開きます。
2. 検索バーに次のリンクを貼り付けて **Enter** を押します。

<Service URL>?NumberA=8&NumberB=0

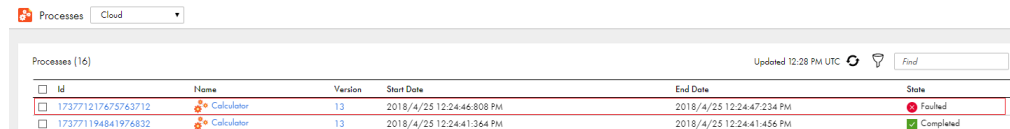
0 による除算はできないため、HTTP 500 エラーが表示されます。加算、減算、および乗算の演算に対する結果もありません。



## フォールトになったプロセスインスタンスのアプリケーション統合コンソールでの表示

1. **【マイサービス】** をクリックし、アプリケーション統合コンソールを選択して、**【更新】** をクリックします。

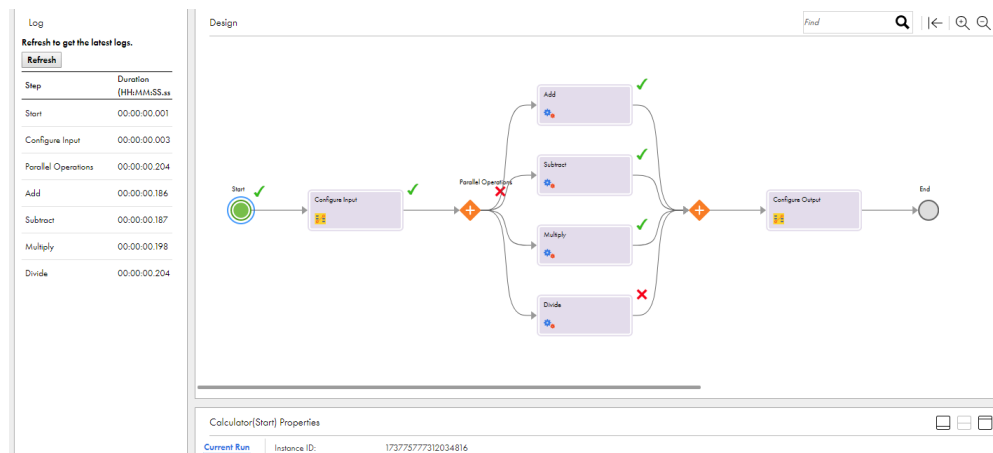
アプリケーション統合コンソールが **【プロセス】** ページに表示されます。電卓プロセスの **【状態】** が **【フォールト】** になっています。



Id	Name	Version	Start Date	End Date	State
173771217675763712	Calculator	13	2018/4/25 12:24:46:808 PM	2018/4/25 12:24:47:234 PM	Faulted
173771194841976832	Calculator	13	2018/4/25 12:24:41:364 PM	2018/4/25 12:24:41:456 PM	Completed

2. プロセスインスタンスの ID をクリックします。

**【プロセスビューの詳細】** が表示されます。**アプリケーション統合コンソール**で、成功したステップとフォールトになったステップにマークが付きます。除算サービスステップでプロセスがフォールトになったことがわかります。



ページ下部のプロパティパネルに詳細を表示するステップを選択します。

## 第 5 章

# フォールト処理を有効化するためのプロセスの更新

プロセスの設計を改善するために、次のタスクを実行します。

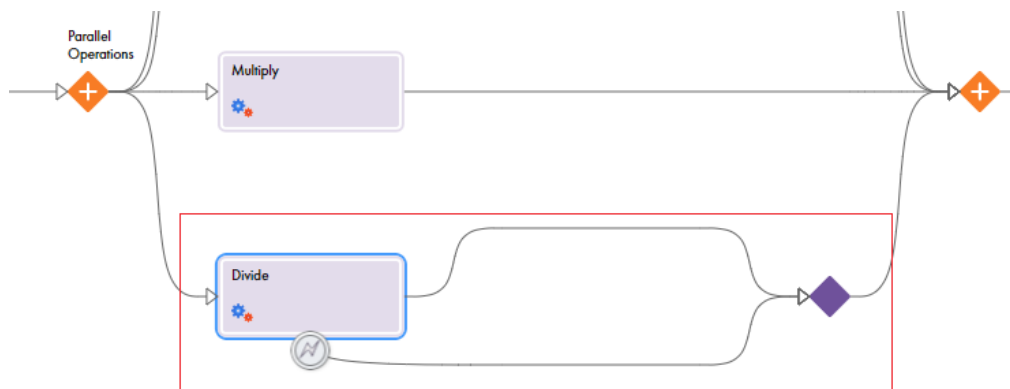
- 除算ステップのフォールト処理を有効化します。除算の結果がエラーの場合も、「HTTPS 500 エラー」の代わりに、「フォールト」などの何らかの値が商に表示される必要があります。
- 除算ができない場合でも、加算、減算、および乗算のステップの出力が表示されるように、プロセスの出力を設定します。

## 手順 1: フォールト処理の有効化

フォールト発生タイマを設定し、フォールトにテキスト値を割り当てることで、除算ステップのフォールト処理を有効にします。

1. アプリケーションの統合にログインし、**【参照】 > 【チュートリアル】 > 【電卓】**の順に移動して、電卓プロセスを開きます。
2. 4 つ目の並列パスにある**【除算】**サービスステップを選択します。  
ステップのプロパティセクションが表示されます。
3. **【フォールト処理】**タブをクリックし、**【フォールトの取得】**を選択します。

**【除算】** サービスステップから 2 つの収束ブランチが表示されます。下のブランチは、フォールトが発生した場合のパスを示します。上のブランチは、フォールトが発生しない場合のパスを示します。



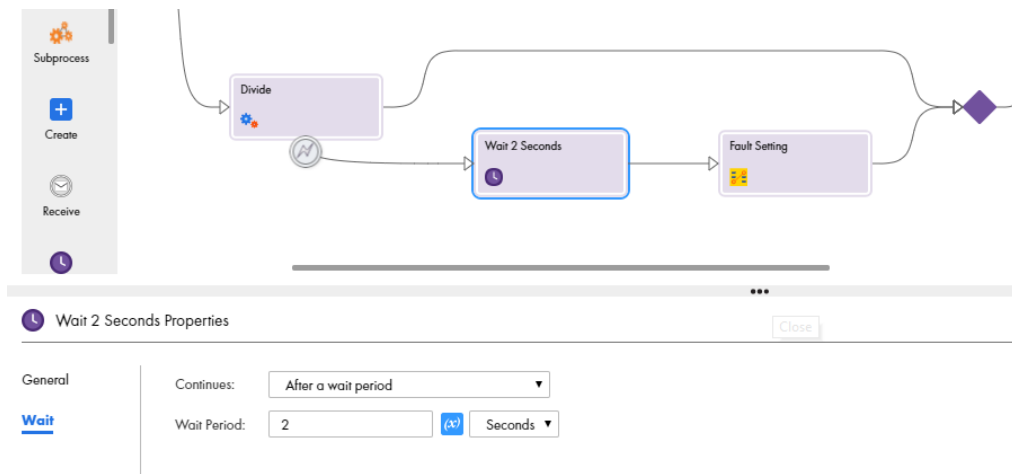
ステップでフォールト処理を有効にした場合は、ステップのスコープ内でフォールト後に発生するイベントを設定します。

4. **【待機】** ステップを **【除算】** サービスステップの下のブランチの上にドラッグし、**【待機】** ステップを選択します。

ステップのプロパティセクションが表示されます。

5. **【待機】** ステップの次のプロパティを設定します。

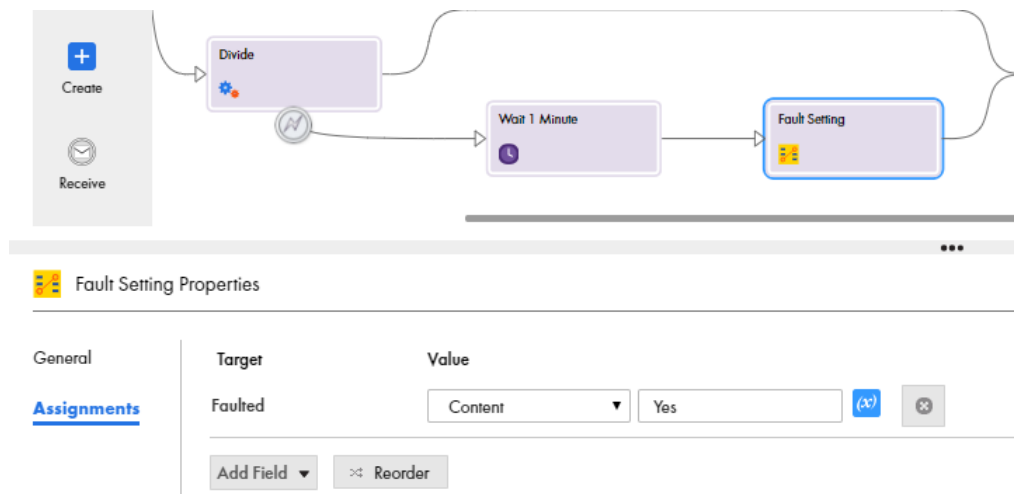
- a. **【全般】** タブを選択し、**【名前】** フィールドに「**2 分待機**」と入力します。
- b. **【待機】** タブを選択し、**【続行】** フィールドを **【待機期間後】** に設定し、**【待機】** フィールドを **【2 秒】** に設定します。



6. **【開始】** ステップを選択し、一時フィールドの名前を **【フォールト】**、タイプを **【テキスト】** にして追加します。詳細については、「プロセスの作成」の「[手順 1: プロセスの作成とプロセスの基本プロパティの設定](#)」の[手順 8](#)を参照してください。

**注:** このフィールドは、プロセスにフォールトがあるかどうかについてディシジョンステップを後で追加するときに使用します。

7. **【割り当て】** ステップを **【除算】** ステップの下ブランチの、**【待機】** ステップの後にドラッグします。
8. **【割り当て】** ステップを選択し、ステップのプロパティセクションで次のプロパティを設定します。
  - a. **【全般】** タブをクリックし、**【名前】** フィールドに「**フォールトの設定**」と入力します。
  - b. **【割り当て】** タブをクリックして、手順 7 で作成した **【フォールト】** ステップを追加します。**【値】** を **【コンテンツ】** > **【はい】** に設定します。



9. **【保存】** をクリックします。

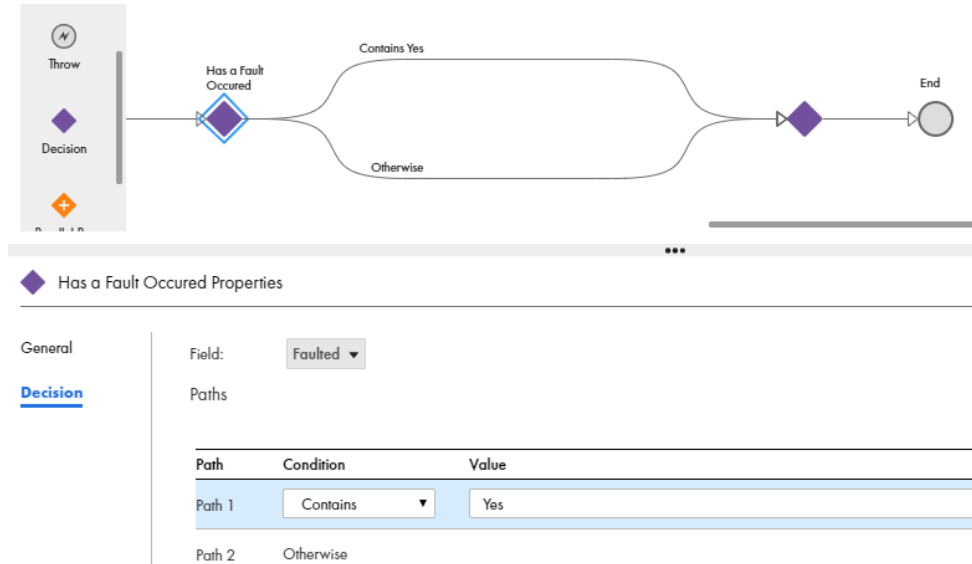
**【除算】** ステップのフォールト処理を有効にしました。また、フォールトが発生した場合にその 2 秒後に値 **【はい】** が割り当てられた **【フォールト】** という一時フィールドも作成しました。

## 手順 2: 出力設定の変更

前提条件: 「プロセスの作成」セクションの「[手順 4: 出力の設定](#)」(ページ 20)で作成した **【出力の設定】** ステップを削除します。出力の設定の **【割り当て】** ステップの代わりに、**【ディシジョン】** ステップを追加します。このステップには、それぞれが **【割り当て】** ステップを持つ 2 つのブランチを設定します。**【割り当て】** ステップを使用して、フォールトのある場合とない場合の両方の出力を割り当てます。

- 1.
2. **【出力の設定】** ステップを選択して **【削除】** ボタンをクリックします。  
**【出力の設定】** ステップがキャンバスに表示されなくなります。
3. **【開始】** ステップを選択し、**【商】** 出力フィールドの **【タイプ】** を **【テキスト】** に変更します。  
これを行うのは、商フィールドの値が、整数ではなく「フォールト」などのテキスト値になる可能性があるためです。
4. **【ディシジョン】** ステップを、キャンバス上の **【終了】** ステップの前にドラッグし、ステップを選択します。  
収束する 2 つのパスを持つ **【ディシジョン】** ステップが表示されます。
5. **【ディシジョン】** ステップの次のプロパティを設定します。
  - a. **【全般】** タブを選択し、**【名前】** フィールドに「**フォールトが発生**」と入力します。
  - b. **【ディシジョン】** タブを選択し、**【フィールド】** リストから **【フォールト】** を選択します。

- c. **【パス 1】** の **【値】** フィールドに、「はい」と入力します。



6. **【割り当て】** ステップを **【次を含む: はい】** と **【Otherwise】** のパスにドラッグします。  
これは、フォールトがある場合でもない場合でも出力を割り当てるために行います。
7. **【次を含む: はい】** パスの **【割り当て】** ステップを選択します。  
ステップのプロパティセクションが表示されます。
8. **【割り当て】** ステップの次のプロパティを設定します。
  - a. **【全般】** タブを選択し、**【名前】** フィールドに「**フォールトの出力の設定**」と入力します。
  - b. **【割り当て】** タブを選択して、以下のフィールドを追加します。

フィールド	タイプ	値
和	フィールド	AddResponse > AddResult
差	フィールド	SubtractResponse > SubtractResult
積	フィールド	MultiplyResponse > MultiplyResult
商	コンテンツ	<b>フォールト</b>

9. **【Otherwise】** パスの **【割り当て】** ステップを選択します。  
ステップのプロパティセクションが表示されます。
10. **【割り当て】** ステップの次のプロパティを設定します。
  - a. **【全般】** タブを選択し、**【名前】** フィールドに「**フォールトがない場合の出力の設定**」と入力します。

- b. **【割り当て】** タブを選択して、以下のフィールドを追加します。

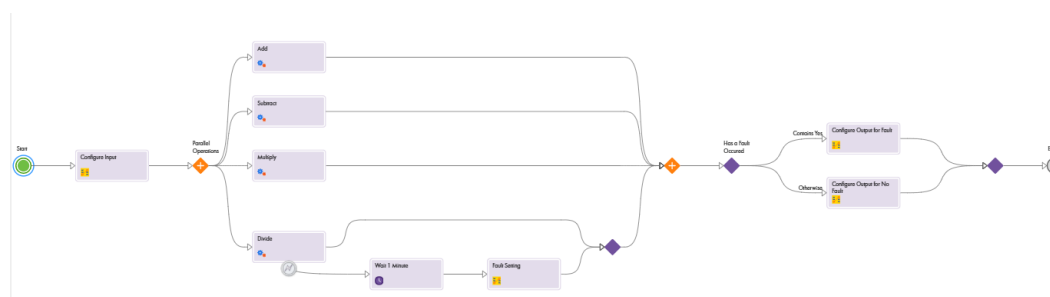
フィールド	タイプ	値
和	フィールド	AddResponse > AddResult
差	フィールド	SubtractResponse > SubtractResult
積	フィールド	MultiplyResponse > MultiplyResult
商	フィールド	DivideResponse > DivideResult

11. **【検証】** をクリックします。エラーがない場合には、**【保存】** をクリックします。

エラーがある場合は、エラーを修正してプロセスを保存します。

12. **【アクション】** > **【パブリッシュ】** をクリックします。

次の図は、変更が完了したプロセスを示しています。



## 第 6 章

# 整数 8 と 0 を使用した更新後のプロセスの起動

変更したプロセスを整数 8 と 0 を使用して起動し、その結果を「異なる整数を使用したプロセスの起動」の[手順 2: 整数 8 と 0 を使用したプロセスの起動 \(ページ 23\)](#)セクションで確認した起動エラーと比較します。

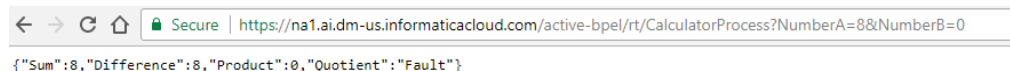
1. ブラウザを開きます。
2. 検索バーに次のリンクを貼り付けて **Enter** を押します。

<Service URL>?NumberA=8&NumberB=0

サービス URL を取得するには、「[手順 1: 整数 10 と 5 を使用したプロセスの起動](#)」の [2](#) を参照してください。

次の値で応答が表示されます。





- 和: 8
- 差: 8
- 積: 0
- 商: フォールト



```
{"Sum":8,"Difference":8,"Product":0,"Quotient":"Fault"}
```

## 更新されたプロセスインスタンスのアプリケーション統合コンソールでの表示

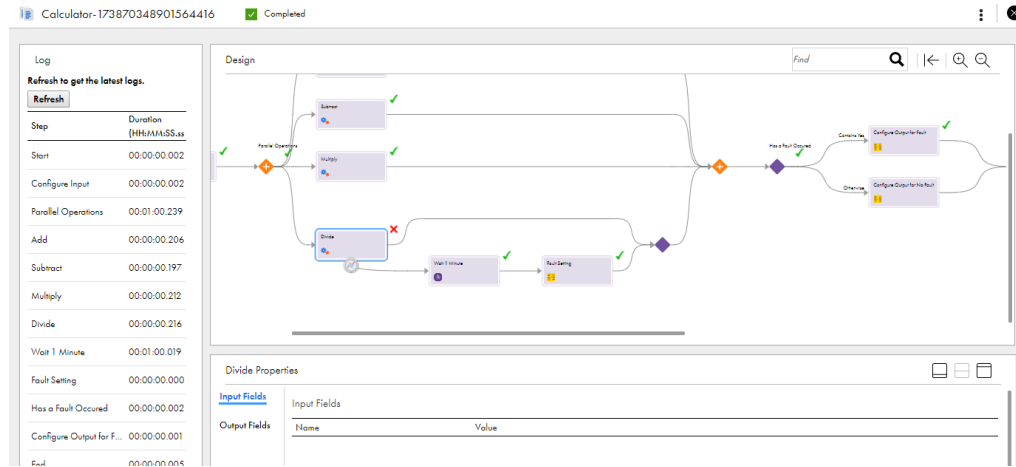
1. **【マイサービス】** をクリックし、アプリケーション統合コンソールを選択して、更新をクリックします。アプリケーション統合コンソールが **【プロセス】** ページに表示されます。電卓プロセスの **【状態】** が **【成功】** になっています。

Processes (25)					
Updated 7:06 PM UTC   Find					
<input type="checkbox"/> Id	Name	Version	Start Date	End Date	State
<input type="checkbox"/> 173870348901564416	 Calculator	1.4	2018/4/25 06:58:41:534 PM	2018/4/25 06:59:41:954 PM	 Completed

**注:** プロセスインスタンスのバージョンが、13 から 14 に上がりました。元のプロセスを実行したとき、プロセスバージョンは 13 でした。変更したプロセスを実行したときに、プロセスバージョンは 14 になりました。

2. プロセスインスタンスの ID をクリックします。

【プロセスビューの詳細】が表示されます。除算サービスステップは失敗しましたが、フォールト処理ブランチの【待機】と【割り当て】ステップが成功して、フォールトが処理されました。



**注:** 除算ステップはフォールトとしてマークされていますが、フォールトが処理されたため、並列パスステップは成功としてマークされています。

## 第 7 章

# オプション: 結果のアーカイブ

電卓サービスプロセスの結果をアーカイブできます。作成ステップで、JDBC 接続を作成して接続を使用します。

## JDBC 接続の作成

接続プロパティを入力して、JDBC 接続を作成します。

**前提条件:** JDBC 接続を作成するには、稼動している Secure Agent と、JDBC データベースへの書き込みアクセスが必要です。

次のコードを使用して、JDBC データベースにスキーマを作成します。

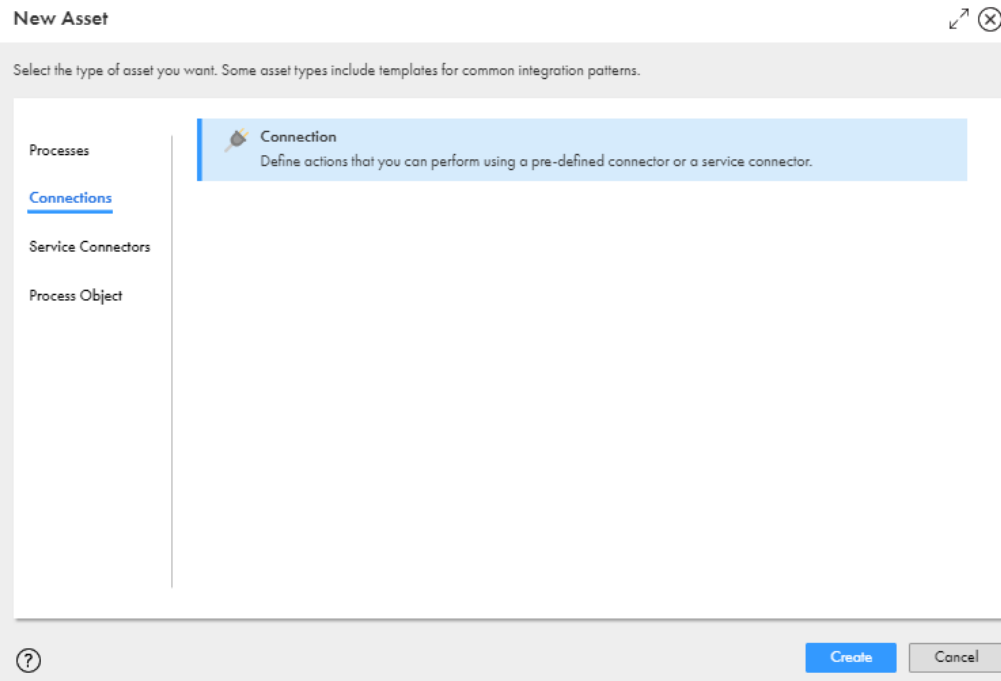
```
CREATE TABLE `calculatorArchive` (  
  `sum` int(11) DEFAULT NULL,  
  `difference` int(11) DEFAULT NULL,  
  `product` int(11) DEFAULT NULL,  
  `quotient` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

次の JDBC プロパティを書き留めておきます。

- 接続 URL
- ユーザー名
- パスワード
- Secure Agent を含むマシンの JDBC JAR ディレクトリ。例えば、Secure Agent が同じマシンの C:\Program Files\InformaticaCloudSecureAgent にある場合、C:\JDBC は有効な JDBC JAR ディレクトリです。

**注:** 汎用の JDBC コネクタがアプリケーションの統合サービスに組み込まれているため、JDBC サービスコネクタを作成する必要はありません。

1. アプリケーションの統合で、**【新規】** をクリックします。
2. **【新しいアセット】** ダイアログボックスで、**【接続】** > **【接続】** の順に選択し、**【作成】** をクリックします。



接続テンプレートが表示された、Process Designer が開きます。

3. [プロパティ] タブに次の情報を入力します。
  - a. **名前** フィールドに、「**JDBCConn**」と入力します。
  - b. **プロジェクト** フィールドの横で、**参照** をクリックしたら、**チュートリアル** > **電卓** の順にフォルダを移動して選択します。
  - c. **説明** フィールドに、「**電卓プロセスの結果をアーカイブするための JDBC 接続**」というテキストを入力します。
  - d. **タイプ** リストで、JDBC 汎用クラウドアダプタを選択します。
  - e. **実行** リストで、Secure Agent を選択します。
  - f. **JDBC 接続 URL**、**JDBC Jar ディレクトリ**、**ユーザー名**、および **パスワード** に値を入力します。
4. **テスト** をクリックします。

接続のテストが正常に行われます。
5. **保存** をクリックします。
6. **アクション** > **パブリッシュ** をクリックします。

**Connection Details**

---

Name:  (Unpublish connection to edit name)

Location:

Description:

Type:

Run On:

Connection Test:

OData-Enabled: ☒ No ☐ Yes

Allowed Roles for OData:

**Connection Properties**

Name	Value	Description
JDBC Connection URL:	<input type="text" value="jdbc:mysql://174.129.15.178:3306/cal"/>	
JDBC Jar Directory:	<input type="text" value="C:\JDBC"/>	
JDBC Driver Class Name:	<input type="text"/>	
Schema:	<input type="text" value="cal"/>	
Username:	<input type="text" value="test"/>	
Password:	<input type="password" value="*****"/>	
Object Filter:	<input type="text"/>	Comma separated list of object names.

詳細については、『[JDBC Connector Guide](#)』を参照してください。

## 結果のアーカイブ

マイルストーンステップと作成ステップを使用して、プロセスの結果をアーカイブします。

マイルストーンステップは、同期応答を送信します。マイルストーンステップの前のすべてのステップが正常に実行されると、アプリケーションの統合は、作成ステップが失敗しても、返された成功の出力を返します。

1. **【マイルストーン】** ステップを、パレットから **【ディシジョン】** ステップと **【終了】** ステップの間のキャンバス上にドラッグします。
  - a. **【マイルストーン】** ステップを選択します。  
ステップのプロパティセクションが表示されます。
  - b. **【全般】** タブをクリックし、**【名前】** フィールドに「**応答の送信**」と入力します。
2. **【作成】** ステップを、パレットから **【マイルストーン】** ステップと **【終了】** ステップの間のキャンバス上にドラッグします。
3. **【作成】** ステップを選択します。  
ステップのプロパティセクションが表示されます。
4. **【全般】** タブをクリックし、**【名前】** フィールドに「**結果のアーカイブ**」と入力します。
5. **【作成】** タブをクリックして、以下のステップを実行します。
  - a. **【接続】** リストで **【JDBCConn】** を選択します。
  - b. **【オブジェクト】** リストで **【CalculatorArchive】** を選択します。

6. 【入力】タブを選択し、次の入力フィールドを追加してそのタイプと値を設定します。

入力フィールド名	値の種別	値
和	フィールド	AddResponse > AddResult
差	フィールド	SubtractResponse > SubtractResult
積	フィールド	MultiplyResponse > MultiplyResult
商	フィールド	DivideResponse > DivideResult

**calculatorArchive Properties**

General

Create

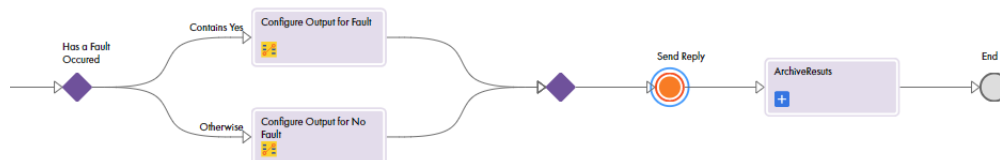
**Input Fields**

Fault Handling

Input Fields (4)

Name	Required	Value
sum	<input type="checkbox"/>	Field AddResponse > AddResult
difference	<input type="checkbox"/>	Field SubtractResponse > SubtractResult
product	<input type="checkbox"/>	Field MultiplyResponse > MultiplyResult
quotient	<input type="checkbox"/>	Field DivideResponse > DivideResult

7. プロセスを保存してパブリッシュします。



次の図は、JDBC データベースのアーカイブ済み出力を示します。

**Result Grid** Filter Rows:

	sum	difference	product	quotient
	18	-2	80	1
	8	8	0	NULL
	15	5	50	2
	8	8	0	NULL
	3	-1	2	0
	43	39	82	20
	6	2	8	2
	4	4	0	NULL
	7	3	10	2
	5	5	0	NULL
	5	5	0	NULL
	8	8	0	NULL
	12	12	0	NULL
	19	-5	84	1
	488	-158	53295	1
	1588	942	408595	4