



Informatica® Cloud Application Integration
November 2023

Tutorial: Calculator

Informatica Cloud Application Integration Tutorial: Calculator
November 2023

© Copyright Informatica LLC 1993, 2023

Publication Date: 2023-11-23

Table of Contents

Preface	4
Chapter 1: Introduction	5
Before you Begin.	6
Concepts, Assets, and Services.	6
Tutorial Objectives.	7
Chapter 2: Create a Service Connector and a Connection	8
Step 1: Import a WSDL File to Create a Service Connector.	8
Service Connector Process Objects.	11
Step 2: Test Actions.	13
Step 3: Create an App Connection.	13
Chapter 3: Create a Process	16
Step 1: Create a Process.	16
Step 2: Configure the Input.	18
Step 3: Configure Services to Run in Parallel.	19
Step 4: Configure the Output.	21
Chapter 4: Invoke the Process	23
Step 1: Invoke the Process with the Integers 10 and 5.	23
View the Process Instance on the Application Integration Console.	24
Step 2: Invoke the Process with the Integers 8 and 0.	25
View the Faulted Process Instance on the Application Integration Console.	25
Chapter 5: Update the Process to Enable Fault Handling	27
Step 1: Enable Fault Handling.	27
Step 2: Modify the Output Configuration.	29
Chapter 6: Invoke the Updated Process with Integers 8 and 0	31
View the Updated Process Instance on the Application Integration Console.	31
Chapter 7: Optional: Archive Results	33
Create a JDBC Connection.	33
Archive Results.	35

Preface

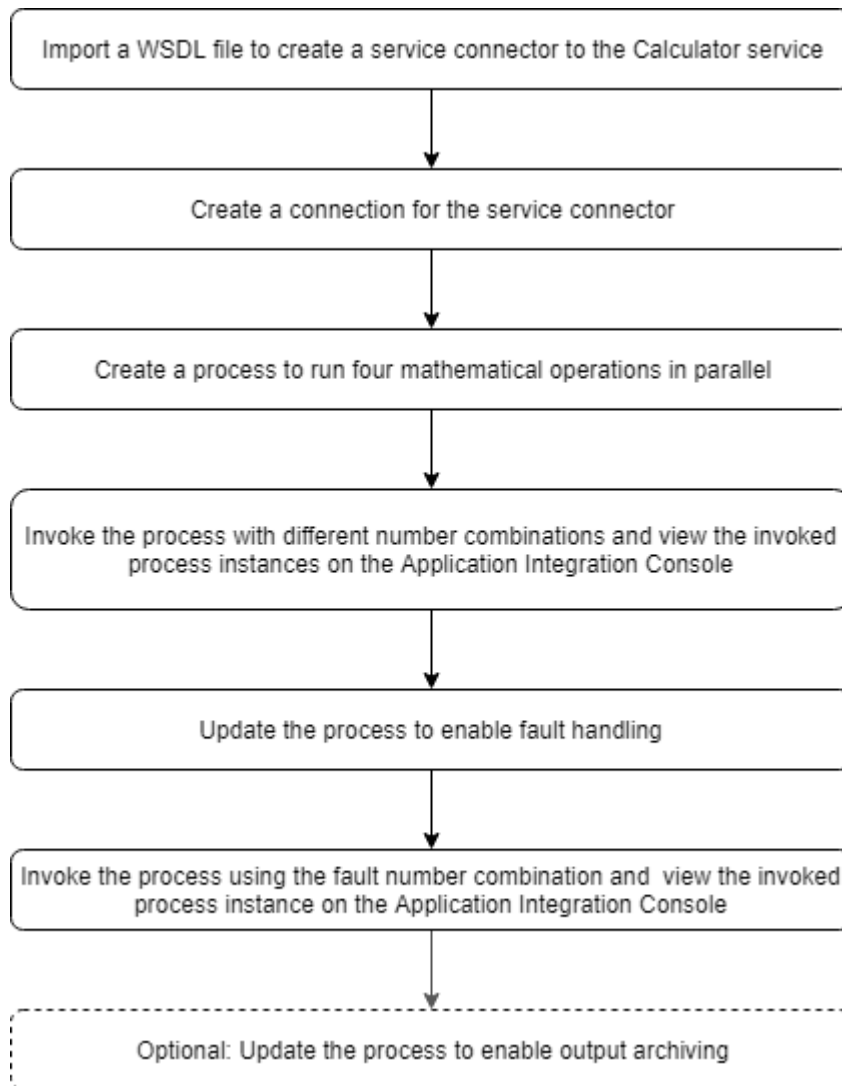
Follow the instructions in the *Calculator* tutorial to learn how to create and invoke a process that runs four operations in parallel.

CHAPTER 1

Introduction

In the *Calculator* tutorial, you use Application Integration to create a process that performs four mathematical operations in parallel. You invoke the process with different integers and enable fault handling when you encounter an error. You also view the invoked process instances on the Application Integration Console.

The following image depicts the process you follow in the *Calculator* tutorial:



Before you Begin

Before you begin, perform the following tasks:

1. Create an Informatica Intelligent Cloud ServicesSM account and verify that you have access to the Application Integration and the Application Integration Console services. For information about creating an account, see the *Register* section of the [Training Program Overview](#) community article.
2. Log in to Application Integration, go to the **Explore** page and select **Projects and Folders** in the **Explore** list.
3. Click **New Project** to create a new **Tutorials** project.
4. Inside **Tutorials**, click **New Folder** to create a **Calculator** folder.

Concepts, Assets, and Services

Learn about the assets that you create and the services and concepts that you use in this tutorial.

Assets

Service Connector

A service connector interacts with services in Informatica Intelligent Cloud Services, within a firewall, or anywhere in the cloud, if the service has an exposed API. In this tutorial, you do not create a service connector from scratch. Instead, you import a WSDL file to create a service connector.

Service Connector Process Object

A process object groups and structures data. A service connector process object groups and structures service connector input and output. The service connector process object is available to any process that uses the service connector. For example, if a service returns demographic information such as name, address, and telephone number, you can create a process object called Demographics to store this output information. You can also create process objects independent of service connectors. This tutorial covers process objects you create within a service connector.

Connection

A connection defines the configuration to connect to a service connector. The service connector may be a built-in connector like JDBC or one that you create, like the Calculator service connector in this tutorial. To use the service connector in a process, you must first create a connection, and then add the connection to the process.

Process

A process is a set of instructions that automate a business requirement. Use process steps to configure a process to run tasks in sequence, obtain information using complex XQuery, take decisions based on some criteria, and much more.

Concepts

Fault Handling

Fault handling refers to configuring the process to account for an exception or an unanticipated condition that occurs during the process flow. With fault handling, a predictable result always occurs. For example, if you think that you might get an error for an output variable, enable fault handling so you see the text "faulted" instead of an HTTPS 500 error.

Services

Application Integration

Application Integration is an Informatica Intelligent Cloud Services service that you can use to perform API-based integration, event processing, and service and process orchestration.

Application Integration Console

Application Integration Console is an Informatica Intelligent Cloud Services service that you can use to perform detailed monitoring and debugging for invoked process instances.

Tutorial Objectives

After you complete this tutorial, you will be able to perform the following tasks:

- Import a WSDL file to create a service connector.
- Create a connection for the service connector.
- Create a process that uses the following steps:
 - Assignment
 - Create
 - Data Decision
 - Parallel Paths
 - Service
- Update the process to enable fault handling.
- View invoked process instances on the Application Integration Console.
- Archive the process output using the Create step. This is an optional task.

CHAPTER 2

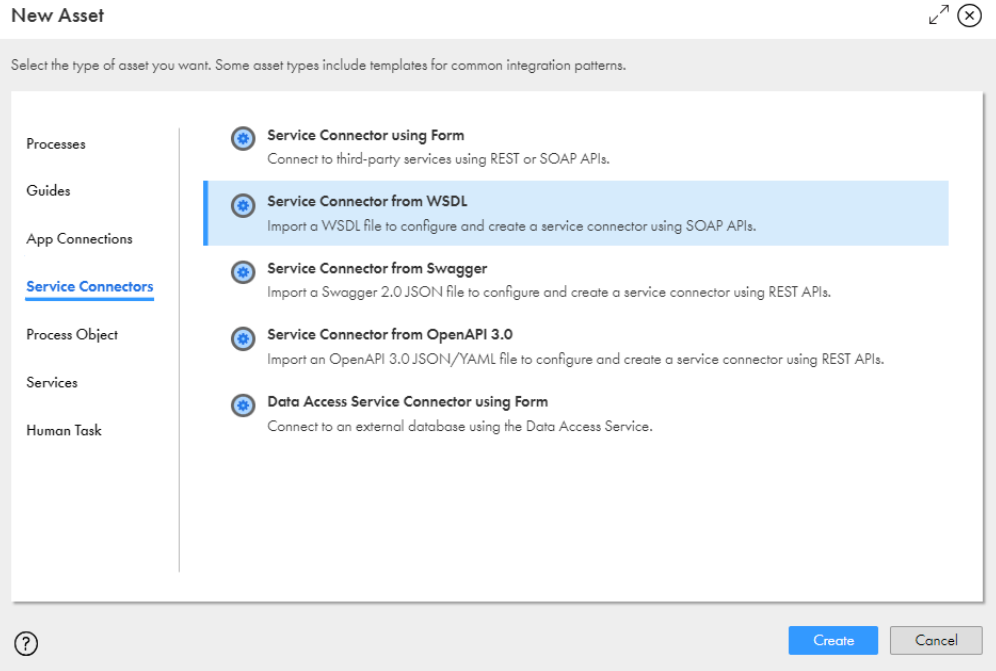
Create a Service Connector and a Connection

Create a service connector to access the addition, subtraction, multiplication, and division services that the <http://www.dneonline.com/calculator.asmx> calculator web service provides. Create the service connector by importing a WSDL file. After you import the WSDL file, you need to perform minor manual tasks to complete the service connector.

After you create the service connector, you must configure a connection to be able to use the service connector in a process.

Step 1: Import a WSDL File to Create a Service Connector

1. Log in to Application Integration and click **New**.
2. In the **New Asset** dialog box, select **Service Connectors** > **Service Connector from WSDL**, and then click **Create** as shown in the following image:



3. In the **New Service Connector from WSDL** dialog box, enter the following properties on the **WSDL Source** tab:
 - a. In the **Name** field, enter **CalculatorSVC**.
 - b. Next to the **Location** field, browse to and select the **Tutorials > Calculator** folder.
 - c. Enter the following text in the **Description** field: **Invoke actions from the dneonline.com/calculator.asmx calculator web service.**
 - d. Next to **WSDL source**, select **URL** and paste the following URL in the **URL** field:
<http://www.dneonline.com/calculator.asmx?WSDL>
 - e. Disable the **Use authentication** option because this service does not require authentication as shown in the following image:

New Service Connector from WSDL

1 WSDL Source 2 Service and Operations 3 Summary 4 Progress

Specify information to identify the Service Connector and WSDL file.

Name: *

Location: *

Description:

svc.import_wizard_wsdl_sourcefile

svc.import_wizard_wsdl_sourcetypes: * File URL

svc.import_wizard_wsdl_sourceurl: *

Use authentication

User Name:

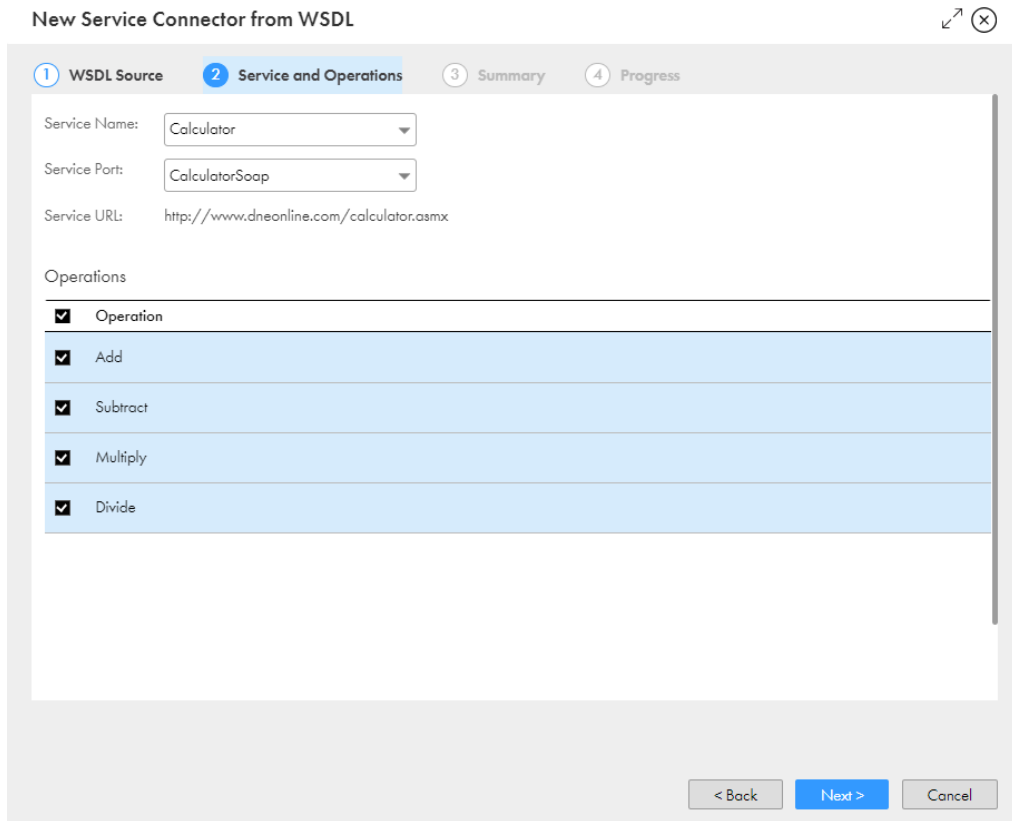
Password:

< Back **Next >** Finish Cancel

4. Click **Next**.

The **Service and Operations** tab opens. You can use the calculator service to perform the following operations: **Add**, **Subtract**, **Multiply**, and **Divide**. Do not change anything on this tab.

The following image shows the **Service and Operations** tab:



5. Click **Next**.

The **Summary** tab appears with details of the service connector and instructions about what you need to do next. Review the service connector configuration.

6. Click **Start Import**.

7. Click **Finish**.

Process Designer opens with the service connector you generated.

Service Connector Process Objects

A service connector process object groups data. The WSDL file you imported to generate a service connector generates eight process objects.

To view the service connector process objects, click the **Process Objects** tab in the **CalculatorSVC** connector.

The following image shows the **Process Objects** tab for the **CalculatorSVC** connector:

▼ Process Objects (8) +

Process Object Name	Description	
AddResponse_AT		✕
Add_AT		✕
DivideResponse_AT		✕
Divide_AT		✕
MultiplyResponse_AT		✕

Properties Fields

Name*:

Description::

The following table describes the service connector process objects:

Name	Description	Number of Fields	Field Names	Field Type
AddResponse_AT	Contains the result of the Add action.	One	AddResult	number
Add_AT	Contains the input required to perform the Add action.	Two	intA, intB	number
DivideResponse_AT	Contains the result of the Divide action.	One	DivideResult	number
Divide_AT	Contains the input required to perform the Divide action.	Two	intA, intB	number
MultiplyResponse_AT	Contains the result of the Multiply action.	One	MultiplyResult	number
Multiply_AT	Contains the input required to perform the Multiply action.	Two	intA, intB	number
SubtractResponse_AT	Contains the result of the Subtract action.	One	SubtractResult	number
Subtract_AT	Contains the input required to perform the Subtract action.	Two	intA, intB	number

Note: The fields contained within the Add_AT, Divide_AT, Multiply_AT, and Subtract_AT process objects are identical. This is because you use the same numbers as input for the Add, Divide, Multiply, and Subtract actions.

Step 2: Test Actions

The WSDL file that you imported contains most of the information necessary to generate a service connector. You need to add test values and test each action.

1. Go to the **Actions** tab and select the **Add** action.
2. Clear the **Preemptive Auth** option.
3. Click the **Input** tab and enter the following data in the **Test With** field:

```
{ "intA": 10, "intB": 2 }
```

After you enter a value in the input field, you must click anywhere outside the row to save the value.

4. Click the **Test Results** tab, and then click **Test**.

The test is successful and you get an output of 12 as shown in the following image:

5. Repeat steps [1](#) through [4](#) for the Subtract, Multiply, and Divide actions.
6. Go to the **Actions > Test Results** tab and click **Test**.
A successful result appears.
7. Click **Save > Publish**.

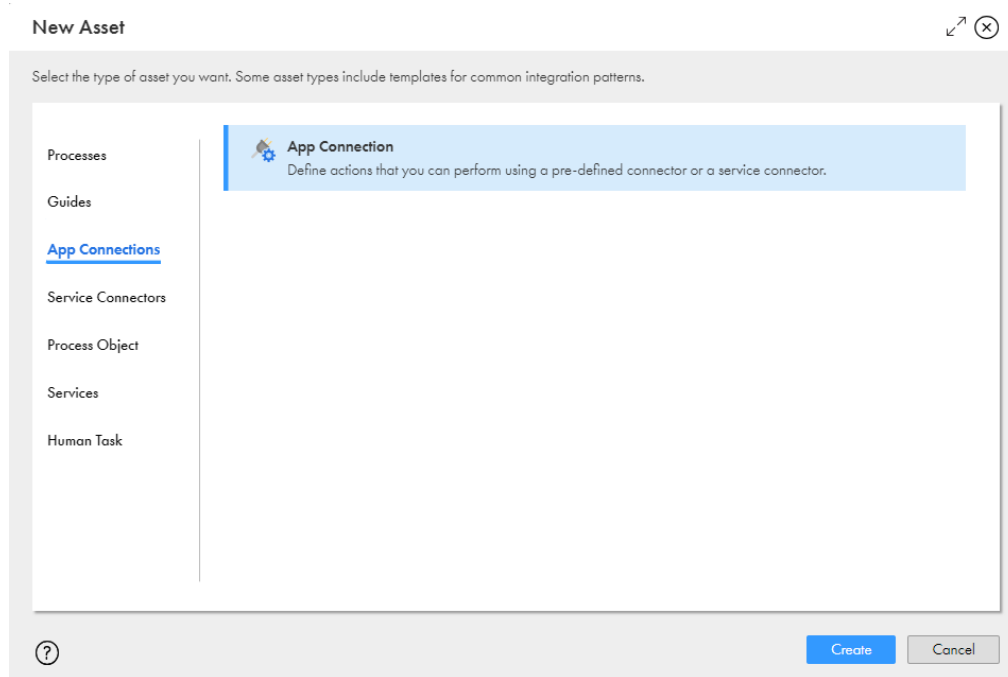
You have created a service connector to access the Calculator service.

Step 3: Create an App Connection

Before you use a service connector in a process, create an app connection for the service connector. Enter connection properties, save the app connection, and publish the connection.

1. In Application Integration, click **New**.

2. In the **New Asset** dialog box, select **App Connections > App Connection**, and then click **Create** as shown in the following image:



Process Designer opens with a connection template.

3. Enter the following information on the **Properties** tab:
 - a. In the **Name** field, enter **CalculatorConn**.
 - b. Next to the **Location** field, click **Browse**, and then browse to and select the **Tutorials > Calculator** folder.
 - c. In the **Description** field, enter the following text: **Use the Add, Subtract, Multiply, and Divide Actions provided by the Calculator service.**
 - d. From the **Type** list, select **Tutorials > Calculator > CalculatorSVC**.

- e. From the **Run On** list, select **Cloud Server or any Secure Agent**.

The following image shows the **Properties** tab in the connection:

CalculatorConn | Valid

[Properties](#) Metadata

Connection Details

Name:* CalculatorConn (Unpublish connection to edit name)

Location: Tutorials\Calculator

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

4. Click **Save** to save the connection.
5. Click **Publish** to publish the connection.

CHAPTER 3

Create a Process

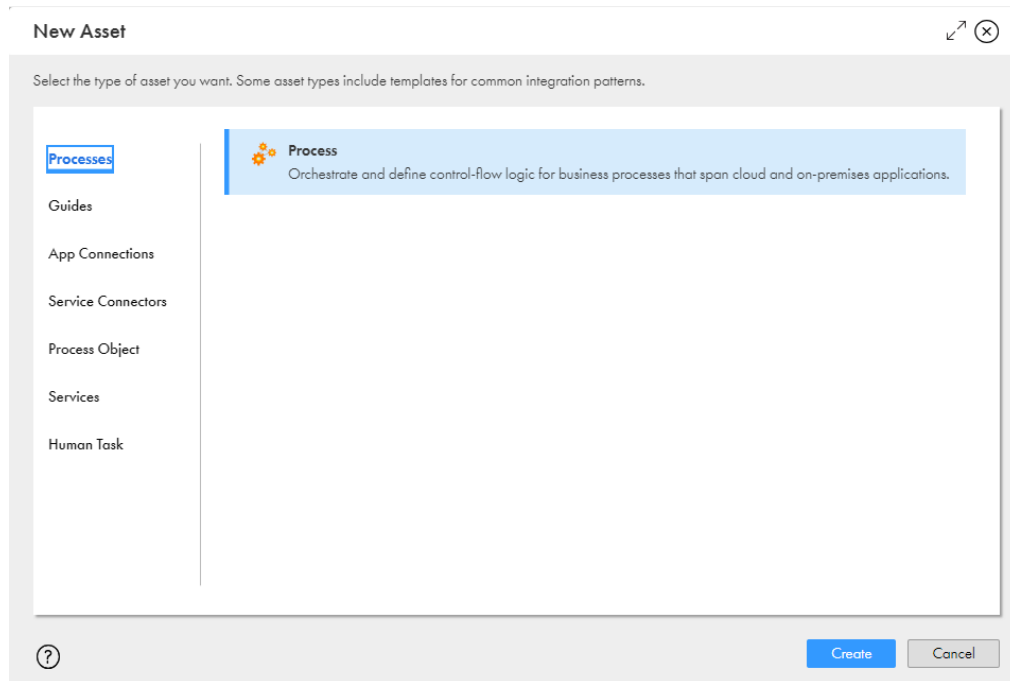
You create a process that performs the four actions provided by the calculator service: Add, Divide, Multiply and Subtract. The process accepts two numbers as input, and then performs each operation on the two numbers in parallel.

Step 1: Create a Process

Create a process and use the **Start** step to configure basic process properties.

1. In Application Integration, click **New**.
2. In the **New Asset** dialog box, select **Processes > Process**, and then click **Create**.

Process Designer opens with a process template as shown in the following image:



3. Click the **Start** step.
The process properties section appears.

4. Click the **General** tab and enter the following properties:
 - a. In the **Name** field, enter **Calculator**.
 - b. Click **Browse** next to **Location**, and then browse to and select the **Tutorials > Calculator** folder.
 - c. In the **Description** field, enter the following text:
Uses the calculator service to perform four mathematical operations in parallel.
5. Click the **Start** tab and enter the following properties:
 - a. From the **Binding** list, select **REST/SOAP**.
 The Binding property defines how a process is invoked.
 - b. Select the **Allow anonymous access** option.
 - c. From the **Run On** list, select **Cloud Server**.
 Do not change the **Applies To** field.
6. Click the **Input Fields** tab, add the following fields, and configure their type:

Name	Type
NumberA	Integer
NumberB	Integer

7. Click the **Output Fields** tab, add the following fields, and configure their type:

Name	Type
Sum	Integer
Difference	Integer
Product	Integer
Quotient	Integer

These fields display the output of each mathematical operation you configure in the process.

8. Click the **Temp Fields** tab, add the following fields, and configure their type:

Name	Type
add_input	More Types > Connection defined Types > Tutorials > Calculator > CalculatorConn > Add_AT
subtract_input	More Types > Connection defined Types > Tutorials > Calculator > CalculatorConn > Subtract_AT

Name	Type
multiply_input	More Types > Connection defined Types > Tutorials > Calculator > CalculatorConn > Multiply_AT
divide_input	More Types > Connection defined Types > Tutorials > Calculator > CalculatorConn > Divide_AT

The service connector expects input in the form of a process object containing two numbers. You define temporary fields to hold the input required for each action.

- Click the **Advanced** tab and set the **Tracing Level** to **Verbose**.

When the tracing level is set to verbose, the Application Integration Console logs all steps. This is useful if you need to debug the process. However, **verbose** logging will increase the process execution time.

Step 2: Configure the Input

Use an Assignment step to define the input fields. You enter input values at runtime when you invoke the process.

- Click and drag an **Assignment** step from the palette on to the canvas in between the **Start** and **End** steps.
- Select the **Assignment** step.
The step properties section appears.
- Click the **General** tab and enter **Configure Input** in the **Name** field.
- Click the **Assignments** tab and add the following fields and configure their type and value:

Field	Type	Value
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

Note: The **Input** fields are available for selection because you defined these fields in *Step 2: Configure Basic Process Properties*.

Provide values for **NumberA** and **NumberB** when you invoke the process.

- Click **Save**.

The following image shows the properties section of the **Configure Input** step:

Field	Assigned Using	From	
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	⬆

You have defined the process input. Next, you must orchestrate the services to run in parallel.

Step 3: Configure Services to Run in Parallel

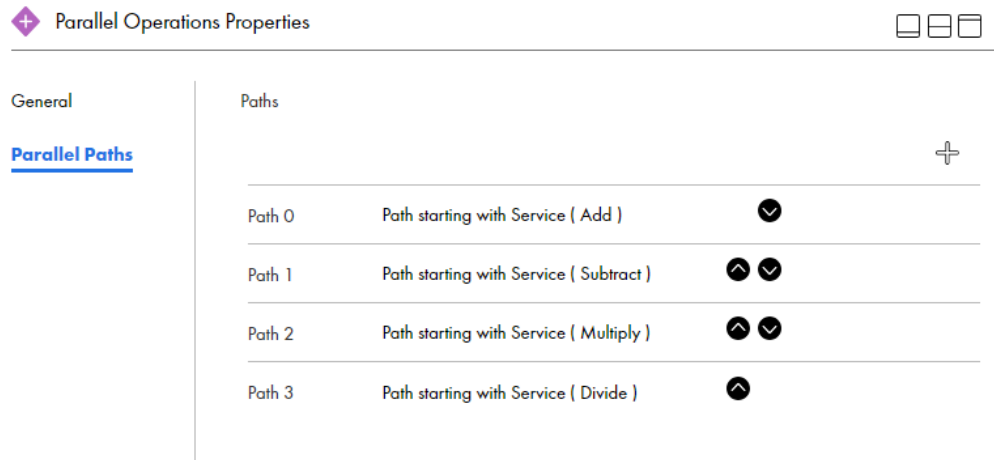
Use the Parallel Paths and Service steps to use the Add, Subtract, Multiply, and Divide operations at the same time.

1. Drag a **Parallel Paths** step from the palette on to the canvas in between the **Assignment** step and **End** steps.
2. Select the **Parallel Paths** step.
The step properties section appears.
3. Click the **General** tab and enter **Parallel Operations** in the **Name** field.
4. Click the **Parallel Paths** tab and perform the following tasks:
 - a. Add four paths.
 - b. Drag a **Service Step** on to each path. Set the **Service Type** to **Connection** and the **Connection** to **Tutorials > Calculator > CalculatorConn**
 - c. Select the **Add** action for the first path.
 - d. Click the **Input Fields** tab, and add the input field **Add**, with type **Field** and value **add_input**.
 - e. Repeat steps 4c and 4d for the three remaining paths.

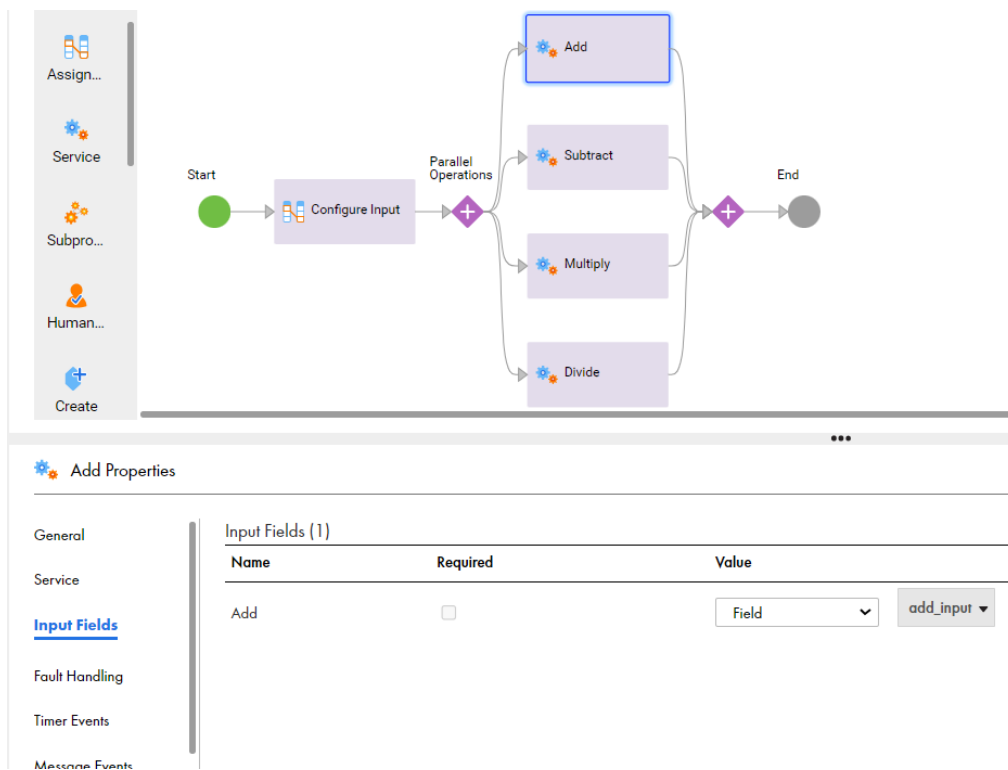
The following table shows the Action that you need to add to the Service step on each path:

Path	Service Type	Connection	Action	Input Field
Path 0	Connection	Tutorials > Calculator > CalculatorConn	Add	Add > Field > add_input
Path 1	Connection	Tutorials > Calculator > CalculatorConn	Subtract	Subtract > Field > subtract_input
Path 2	Connection	Tutorials > Calculator > CalculatorConn	Multiply	Multiply > Field > multiply_input
Path 3	Connection	Tutorials > Calculator > CalculatorConn	Divide	Divide > Field > divide_input

The following image shows the step properties section of the **Parallel Paths** step:



The following image shows the properties section of the **Add** service step:



5. Click **Save**.

Step 4: Configure the Output

Use an Assignment step to define the output that you want to see when you invoke the process. You need output fields for the addition, subtraction, multiplication, and division services.

1. Click and drag an **Assignment** step from the palette on to the canvas in between the **Parallel Paths** and **End** steps.
2. Select the **Assignment** step.
The step properties section appears.
3. Click the **General** tab and enter **Configure Output** in the **Name** field.
4. Click the **Assignment** tab and add the following fields:

Field	Type	Value
Sum	Field	AddResponse > AddResult
Difference	Field	SubtractResponse > SubtractResult

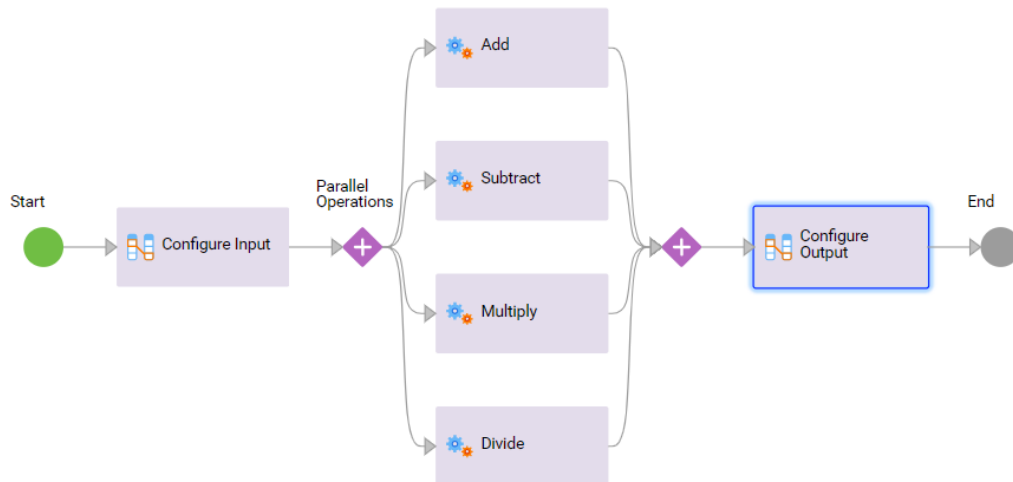
Field	Type	Value
Product	Field	MultiplyResponse > MultiplyResult
Quotient	Field	DivideResponse > DivideResult

5. Click **Save**.
6. Click the **Validation** button. If there are no errors, click **Save**.
If there are errors, fix them, and then save the process.
7. Click **Publish**.

The following image shows the properties section of the **Configure Output** step:

Field	Assigned Using	From	Status
Sum	Field	AddResponse > AddResult	✓
Difference	Field	SubtractResponse > SubtractResult	⬆️ ✓
Product	Field	MultiplyResponse > MultiplyResult	⬆️ ✓
Quotient	Field	DivideResponse > DivideResult	⬆️

The following image shows the complete process:



CHAPTER 4

Invoke the Process

Invoke the process with two sets of integers: 10, 5 and 8, 0. You can view the output you get and view the process instances on the Application Integration Console.

Step 1: Invoke the Process with the Integers 10 and 5

Use a browser to invoke the process service URL with the integers 10 and 5. You see a successful process invocation.

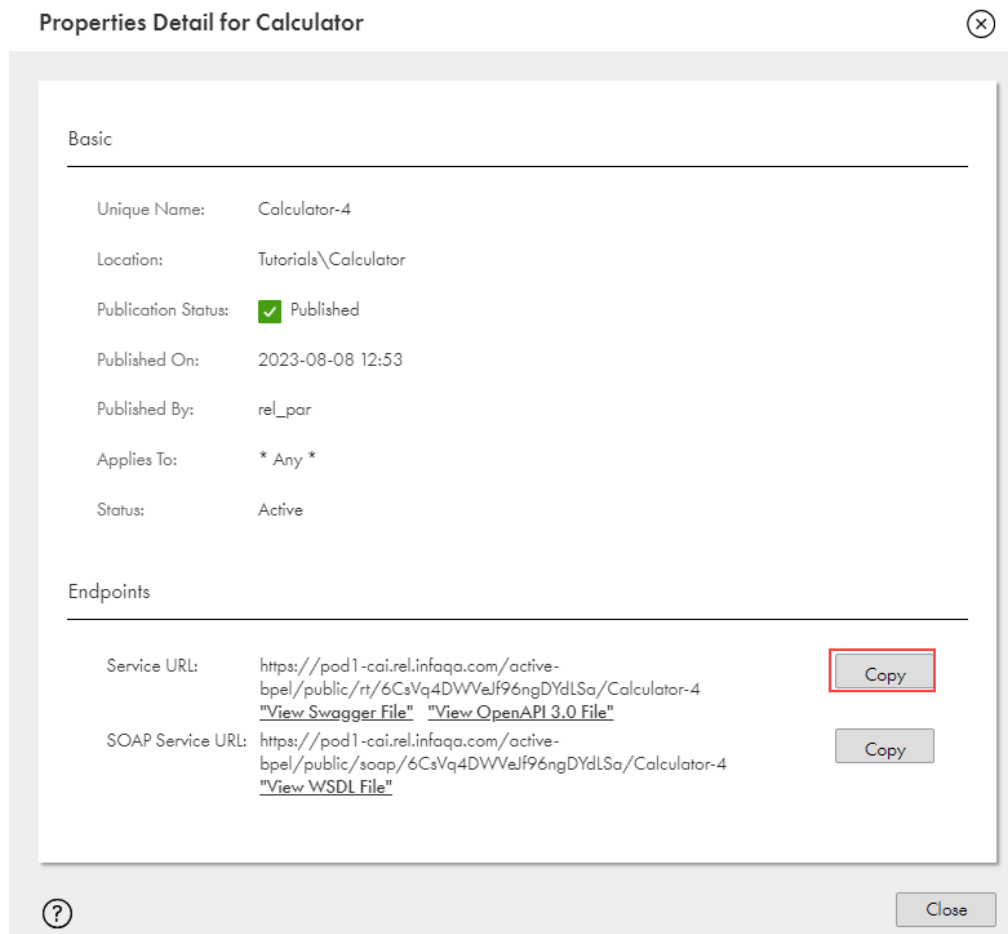
1. In Application Integration, click **Actions > Properties Detail** next to the process.

You can do this from Process Designer or from the **Explore** page.

The **Properties Detail for Calculator** window appears.

2. Copy the Service URL.

The following image shows the **Copy** option on the **Properties Detail for Calculator** window:



3. Open a browser, paste the following link in the search bar, and click **Enter**:

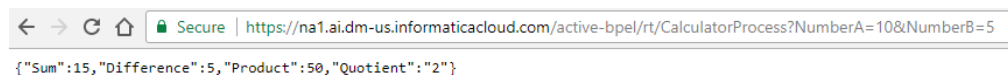
<Service URL>?NumberA=10&NumberB=5

For example, paste the following URL:

<https://na1.ai.dm-us.informaticacloud.com/active-bpel/rt/CalculatorProcess?NumberA=10&NumberB=5>

You see a response with the following values:

- Sum: 15
- Difference: 5
- Product: 50
- Quotient: 2



View the Process Instance on the Application Integration Console

Click the **My Services** list and select Application Integration Console.

The Application Integration Console opens on the **Processes** page. The **Status** of the Calculator process is **Completed** as shown in the following image:

Processes Cloud

Processes (1 item found) Updated 12:59 PM IST / 7:29 AM UTC

ID	Name	Version	Start Date	End Date	Status
873467243160563712	Calculator	1	2023/8/8 07:26:59:010 AM	2023/8/8 07:27:02:139 AM	Completed

Step 2: Invoke the Process with the Integers 8 and 0

Invoke the process service URL with the integers 8 and 0. You will see an error because division by zero is not possible.

1. Open a browser.
2. Paste the following link in the search bar and click **Enter**:

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

You see an HTTP 500 error because division by zero is not possible. There are no results for the Add, Subtract, and Multiply operations either.

```

Secure | https://na1.ai.dm-us.informaticacloud.com/active-bpel/rt/CalculatorProcess-17?NumberA=8&NumberB=0
{"error":{"code":500,"detail":{"reason":{"soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" <soap:Body> <soap:Fault> <faultcode>soap:Server <faultcode> <faultstring>System.Web.Services.Prot unable to process request. ---> System.OverflowException: Arithmetic operation resulted in an overflow. \n at Calculator.Divide(Int32 IntA, Int32 IntB) \n --- End of inner exception str <detail/> \n </soap:Fault> \n </soap:Body> \n </soap:Envelope> \n"}, "code": "HTTP_500"}, "message": {"soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" <soap:Body> <faultcode>soap: <faultstring>System.Web.Services.Protocols.SoapException: Server was unable to process request. ---> System.OverflowException: Arithmetic operation resulted in an overflow. \n at Calcul intB) \n --- End of inner exception stack trace ---</faultstring> \n <detail/> \n </soap:Fault> \n </soap:Body> \n </soap:Envelope> \n"}

```

View the Faulted Process Instance on the Application Integration Console

1. Click **My Services**, select Application Integration Console, and then click **Refresh**.

The Application Integration Console opens on the **Processes** page. The **Status** of the Calculator process is **Faulted**.

Processes Cloud

Processes (2 items found) Updated 1:01 PM IST / 7:31 AM UTC

ID	Name	Version	Start Date	End Date	Status
873468288876048384	Calculator	1	2023/8/8 07:31:08:326 AM	2023/8/8 07:31:08:493 AM	Faulted
873467243160563712	Calculator	1	2023/8/8 07:26:59:010 AM	2023/8/8 07:27:02:139 AM	Completed

2. Click the process instance ID.

The **Process View Detail** appears. **Application Integration Console** marks successful and failed steps. You can see that the process faulted at the Divide Service step.

Log

Refresh to get the latest logs.

Refresh

Step	Duration [HH:MM:SS.ss]
Start	00:00:00.001
Configure Input	00:00:00.003
Parallel Operations	00:00:00.204
Add	00:00:00.186
Subtract	00:00:00.187
Multiply	00:00:00.198
Divide	00:00:00.204

Design

Start

Configure Input

Parallel Operations

Add

Subtract

Multiply

Divide

Configure Output

End

Calculator(Start) Properties

Current Run Instance ID: 173775777312034816

Select a step to view the details in the properties panel at the bottom of the page.

CHAPTER 5

Update the Process to Enable Fault Handling

Perform the following tasks to improve the process design:

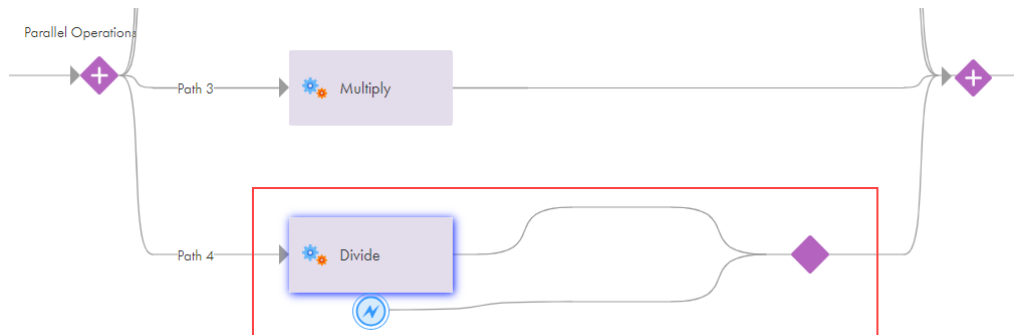
- Enable fault handling for the Divide step. Even if division results in an error, some value, for example, "faulted," must appear for the quotient instead of an HTTPS 500 error.
- Configure the process output such that even if division is not possible, the output of the Add, Subtract, and Multiply steps appear.

Step 1: Enable Fault Handling

Enable fault handling for the Division step by setting a timer for when a fault occurs and assigning a text value to the fault.

1. Log in to Application Integration, go to **Explore > Tutorials > Calculator** and open the Calculator process.
2. Select the **Divide** Service step on the fourth parallel path.
The step properties section appears.
3. Click the **Fault Handling** tab and select **Catch Faults**.

Two converging branches emerge from the **Divide** Service step. The lower branch represents the path taken if a fault occurs. The upper branch represents the path taken if no fault occurs as shown in the following image:

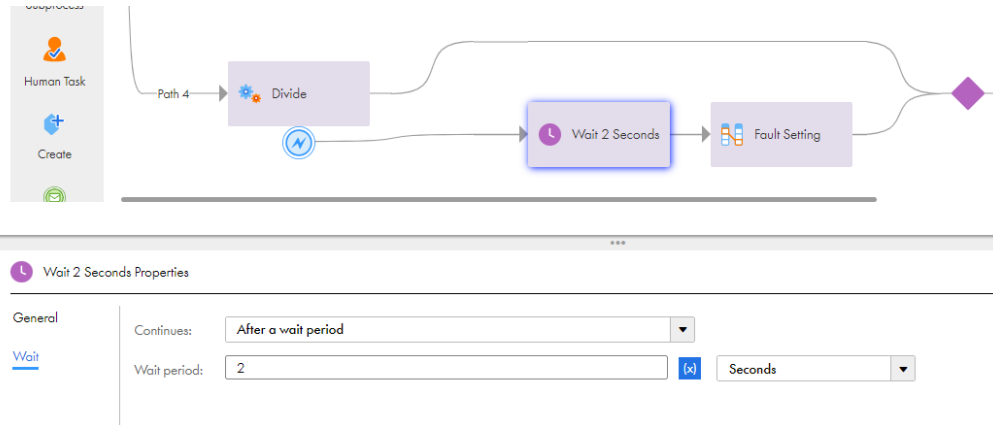


When you enable fault handling for a step, you configure an event that happens after the fault but within the scope of the step.

4. Drag a **Wait** step on to the lower branch of the **Divide** Service step and select the **Wait** step.

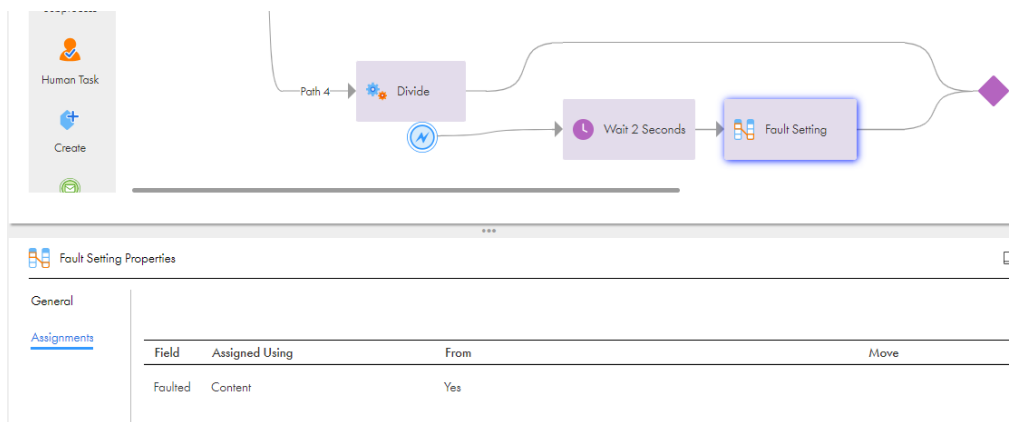
The step properties section appears.

5. Configure the following properties for the **Wait** Step:
 - a. Click the **General** tab and enter **Wait 2 Seconds** in the **Name** field.
 - b. Click the **Wait** tab and set the **Continues** field to **After a wait period** and set the **Wait** field to **2 Seconds** as shown in the following image:



6. Click the **Start** step and add a temporary field with name **Faulted** and Type **Text**. For more information, see step 8 in *Step 1: Create a Process and Configure Basic Process Properties* under *Create a Process*.

Note: You use this field when you later add a decision step about whether the process has a fault or not.
7. Drag an **Assignment** step onto the lower branch of the **Divide** step just after the **Wait** step.
8. Select the **Assignment** step and configure the following properties in the step properties section:
 - a. Click the **General** tab and enter **Fault Setting** in the **Name** field.
 - b. Click the **Assignments** tab, add the **Faulted** field that you created in step 6. Set the **Value** to **Content > Yes** as shown in the following image:



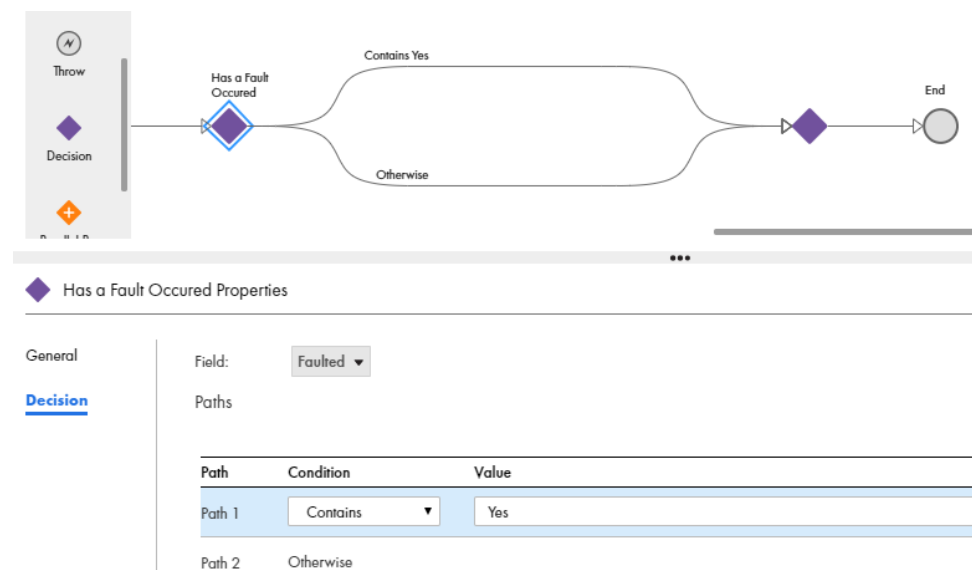
9. Click **Save**.

You have enabled fault handling for the **Divide** step. You have also created a temporary field called **Faulted** that is assigned the value **Yes** two seconds after a fault occurs, if it occurs.

Step 2: Modify the Output Configuration

Prerequisite: Delete the **Configure Output** step that you created in “[Step 4: Configure the Output](#)” on page 21 of the *Create a Process* section. In place of the Configure Output Assignment step, add a Decision step with two branches, each with an Assignment step. Use the Assignments steps to assign outputs for both a fault scenario and a non-fault scenario.

1. Select the **Configure Output** step and click the **Delete** button.
The **Configure Output** step disappears from the canvas.
2. Select the **Start** step and change the **Type** of the **Quotient** output field to **Text**.
You do this because Quotient might have a text value, of say, "fault" instead of an integer.
3. Drag a **Decision** step on to the canvas before the **End** step and select the step.
A Decision step with two converging paths appears.
4. Configure the following properties for the **Decision** step:
 - a. Click the **General** tab and enter **Has a Fault Occurred** in the **Name** field.
 - b. Click the **Decision** tab and select **Faulted** from the **Field** list.
 - c. In the **Value** field of **Path 1**, enter **Yes** as shown in the following image:



5. Drag Assignment steps onto the **Contains Yes** and **Otherwise** paths.
You do this to assign outputs in the case of a fault and if there is no fault.
6. Select the **Assignment** step on the **Contains Yes** path.
The step properties section appears.
7. Configure the following properties for the **Assignment** Step:
 - a. Click the **General** tab and enter **Configure Output for Fault** in the **Name** field.

- b. Click the **Assignments** tab and add the following fields:

Field	Type	Value
Sum	Field	AddResponse > AddResult
Difference	Field	SubtractResponse > SubtractResult
Product	Field	MultiplyResponse > MultiplyResult
Quotient	Content	Fault

8. Select the **Assignment** step on the **Otherwise** path.

The step properties section appears.

9. Configure the following properties for the **Assignment** Step:

- Click the **General** tab and enter **Configure Output for No Fault** in the **Name** field.
- Click the **Assignments** tab and add the following fields:

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

10. Click **Validation**. If there are no errors, click **Save**.

If there are errors, fix the errors, and then save the process.

11. Click **Actions > Publish**.

The following image shows the complete modified process:



CHAPTER 6

Invoke the Updated Process with Integers 8 and 0

Invoke the modified process with the integers 8 and 0 and compare it with the invoke error you saw in the [Step 2: Invoke the Process with the Integers Eight and Zeroon page 25](#) section of *Invoke the Process with Different Integers*.

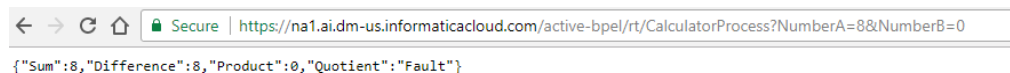
1. Open a browser.
2. Paste the following link in the search bar and click **Enter**:

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

To get the Service URL, see [2](#) of *Step 1: Invoke the Process with the Integers 10 and 5*

You see a response with the following values:

- Sum: 8
- Difference: 8
- Product: 0
- Quotient: Fault



Secure | <https://na1.ai.dm-us.informaticacloud.com/active-bpel/rt/CalculatorProcess?NumberA=8&NumberB=0>

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

View the Updated Process Instance on the Application Integration Console

1. Click **My Services**, select Application Integration Console, and then click refresh.

The Application Integration Console opens on the **Processes** page. The **State** of the Calculator process is **Completed**.

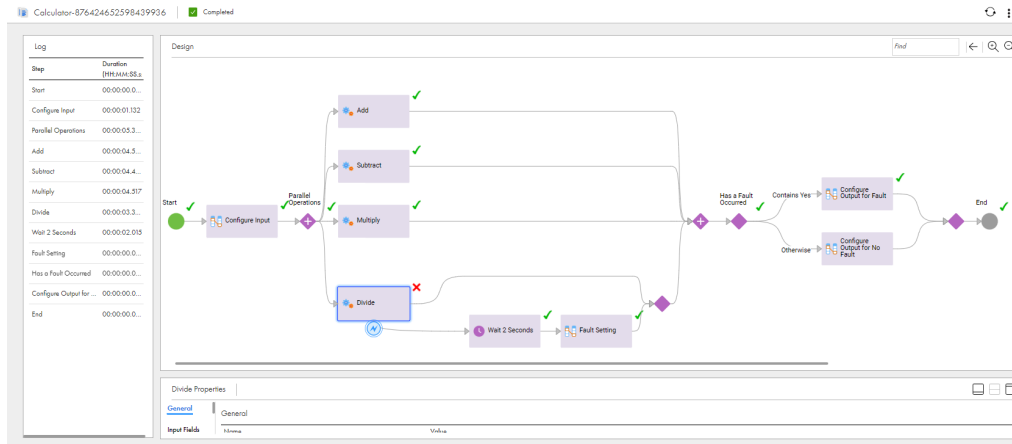


Id	Name	Version	Start Date	End Date	State
173870348901564416	Calculator	14	2018/4/25 06:58:41:534 PM	2018/4/25 06:59:41:954 PM	Completed

Note: The process instance version has incremented, for example, from 13 to 14. When you ran the original process, it was process version 13. When you ran the modified process, it became process version 14.

2. Click the process instance ID.

The **Process View Detail** appears. The Divide Service step failed, but the fault was handled by the successful Wait and Assignment steps on the fault handling branch.



Note: Even though the Divide step is marked as faulted, the Parallel Paths step is marked as successful because the fault was handled.

CHAPTER 7

Optional: Archive Results

You can archive the results of the calculator service process. Create a JDBC connection and use the connection in a Create step.

Create a JDBC Connection

Enter connection properties to create a JDBC connection.

Prerequisite: To create a JDBC connection, you must have a working Secure Agent and write access to a JDBC database.

Use the following code to create a schema in your JDBC database:

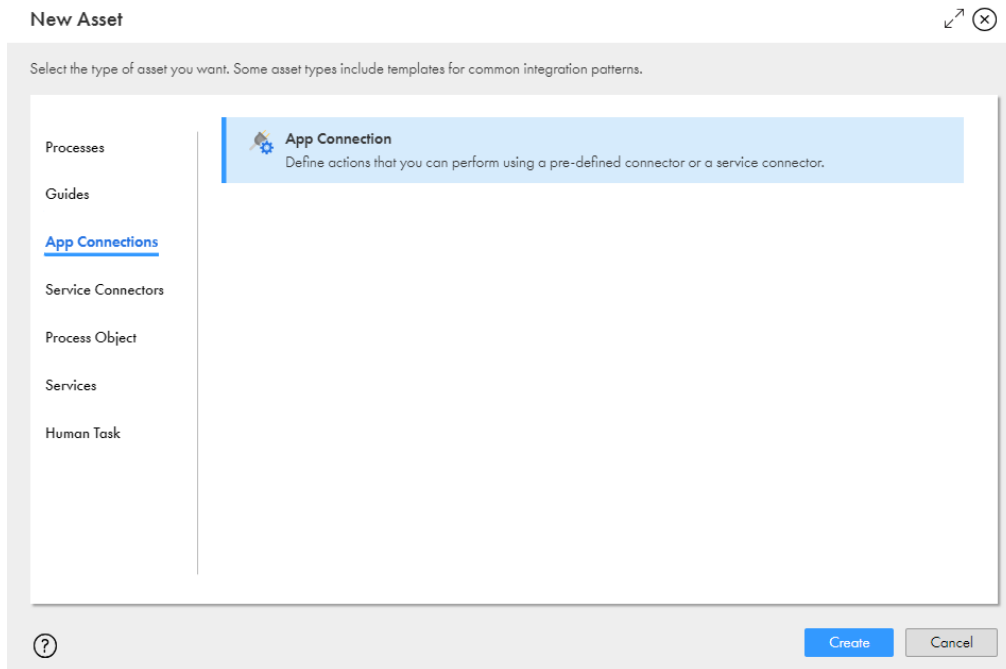
```
CREATE TABLE `calculatorArchive` (  
  `sum` int(11) DEFAULT NULL,  
  `difference` int(11) DEFAULT NULL,  
  `product` int(11) DEFAULT NULL,  
  `quotient` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Keep the following JDBC properties handy:

- Connection URL
- User name
- Password
- JDBC JAR Directory on the machine where you installed the Secure Agent. For example: C:\JDBC is a valid JDBC JAR Directory because the Secure Agent is on the same machine at C:\Program Files\InformaticaCloudSecureAgent

Note: You do not need to create a JDBC service connector because a generic JDBC connector is built in the Application Integration service.

1. In Application Integration, click **New**.
2. In the **New Asset** dialog box, select **App Connections > App Connection**, and then click **Create** as shown in the following image:



Process Designer opens with a connection template.

3. Enter the following information on the **Properties** tab:
 - a. In the **Name** field, enter **JDBConn**.
 - b. Next to the **Location** field, click **Browse**, and then browse to and select the **Tutorials > Calculator** folder.
 - c. In the **Description** field, enter the following text: **JDBC connection to archive Calculator process results**.
 - d. From the **Type** list, select **JDBC Generic Cloud Adaptor**.
 - e. From the **Run On** list, select your Secure Agent.
 - f. Configure the **JDBC Connection URL**, **JDBC Jar Directory**, **Username**, and **Password** fields:
4. Click **Test**.

The connection tests successfully.
5. Click **Save**.
6. Click **Actions > Publish**.

The following image shows the **Properties** page.

Connection Details

Name: *

Location: *

Description:

Type: *

Run On: *

Connection Test:

OData-Enabled: No Yes

OData Cloud Access Enabled: No Yes

Allowed Users for OData:

Allowed Groups for OData:

Connection Properties:

Name	Value	Description
JDBC Connection URL *	jdbc:mysql://174.129.15.178:3306/cal	
JDBC Jar Directory	C:\JDBC	
JDBC Driver Class Name		
Schema	cal	
Username *	test	
Password *	
Object Filter		Comma separated list of object names.
Time Zone	Default	Database timezone. Use this property if DB and Agent are in different timezones.

For more information about JDBC connections, see *JDBC Connector Guide*.

Archive Results

Use a Milestone step and a Create step to archive process results.

A Milestone step sends a synchronous reply. If all the steps before the Milestone steps run successfully, Application Integration returns a successful output even if the **Create** step fails.

1. Drag a **Milestone** step from the palette on to the canvas in between the **Decision** and **End** steps.
 - a. Select the **Milestone** step.
The step properties section appears.
 - b. Click the **General** tab and enter **Send Reply** in the **Name** field.
2. Drag a **Create** step from the palette on to the canvas in between the **Milestone** and **End** steps.
3. Select the **Create** step.
The step properties section appears.
4. Click the **General** tab and enter **Archive Results** in the **Name** field.
5. Click the **Create** tab and perform the following steps:
 - a. From the **Connection** list, select **JBCCConn**.
 - b. From the **Object** list, select **CalculatorArchive**.

6. Click the **Input Fields** tab and add the following input fields and configure their type and value:

Input Field Name	Value Type	Value
sum	Field	AddResponse > AddResult
difference	Field	SubtractResponse > SubtractResult
product	Field	MultiplyResponse > MultiplyResult
quotient	Field	DivideResponse > DivideResult

calculatorArchive Properties

General

Create

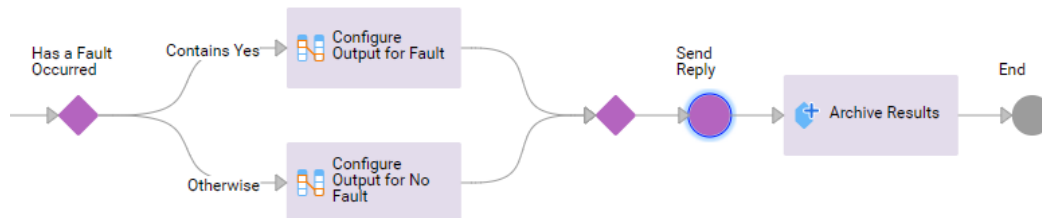
Input Fields

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. Save and publish the process.



The following image shows the archived output in the JDBC database:

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