Integration Guide | PUBLIC
Document Version: 1.1.0 – 2026-05-03

SAP IBP Application Jobs - Reusable Integration Flows

Integrating with SAP IBP Application Jobs

Content

1 Document History

The following table provides an overview of the most important changes:

Version	Date	Description
1.1.0	May 3, 2026	Added information about the following: • Setting up access policies • JMS- and Data Store-based scheduling possibilities.
1.0.0	February 1, 2026	Initial version

2 Introduction

The following integration flows help you connect to application jobs in SAP Integrated Business Planning (SAP IBP):

- *SAP IBP Application Jobs – Schedule Sender*
- *SAP IBP Application Jobs - Schedule with JMS*
- *SAP IBP Application Jobs - Schedule without JMS*
- *SAP IBP Application Jobs - Receiver*
- *SAP IBP Application Jobs – Schedule Receiver without JMS*

There are two scheduling mechanisms available. We recommend the more durable and scalable scheduling with JMS (Java Message Service), for which you need to deploy the following integration flows:

- *SAP IBP Application Jobs – Schedule Sender*
- *SAP IBP Application Jobs - Schedule with JMS*
- *SAP IBP Application Jobs – Schedule Receiver*

As an alternative, a Data Store-based scheduling is also available with similar core functionalities such as asynchronous message handling, but without the more robust performance and reliability benefits of JMS. The throughput and concurrency of scheduled integration flows are determined by the number of runtime nodes of the SAP Cloud Integration tenant. To use this solution, you need to deploy the following integration flows:

- *SAP IBP Application Jobs – Schedule Sender*
- *SAP IBP Application Jobs - Schedule without JMS*
- *SAP IBP Application Jobs – Schedule Receiver without JMS*

The application job template is called *Data Integration Using SAP Cloud Integration*.

Both integration flows need to be deployed for the application jobs to function.

You can find here the required steps in SAP Could Integration to set up the authorization and authentication as well as a description of how you can make your own integration flows conform to the application job templates interfaces.

For more information, see JMS Adapter.

3 Prerequisites

Depending on whether the JMS capability is available, the prerequisites are different.

- For JMS-based scheduling, the prerequisites are the following:
  - Usage of Java Message Service (JMS) is enabled as a capability. For more information, see Activating and Managing Capabilities.
  - The externalized header parameter `SchedulingMode` of the *SAP IBP Application Jobs – Schedule Sender* integration flow must either be set to value 'JMS' or left empty.
  - The target integration flows must expose *Process Direct* inbound addresses.
- For Data Store-based scheduling, the prerequisites are the following:
  - JMS is not required.
  - The externalized header parameter `SchedulingMode` of the *SAP IBP Application Jobs – Schedule Sender* integration flow mustn't be empty and must be set to a value other than 'JMS'.
  - The target integration flows must expose *Process Direct* inbound addresses.

Follow the detailed setup guide in SAP Note 3687976 for scheduling integration flows in SAP Cloud Integration from SAP IBP.

4 Authorization

In SAP Cloud Integration, **inbound** refers to the communication direction when a sender system sends a message to the integration platform. The sender in this case is the SAP IBP application job scheduler. Its authorization check is role-based: the *ESBMessaging.send* predefined role or a custom role including similar authorizations must be assigned to your user.

To configure the deployment of the *SAP IBP Application Jobs – Schedule Sender iFlow*, you need the external parameter *User Role* with your custom created role.

For more information about creating a custom role, see Managing User Roles.

5 Authentication

To establish a connection with SAP IBP, your BTP subaccount administrator must create two instances and their respective service keys.

Creating Instances

To create the first instance for the HTTP adapter in the SAP BTP Cockpit, do the following:

1. Access the sub-account used for SAP Cloud Integration.
2. Go to the *Instances and Subscriptions* tab.
3. Create a new instance and add the following information in the *Basic Info* step:
   - *Service*: Process Integration Runtime (it-rt)
   - *Plan*: integration-flow (Process Integration Flow access)
   - *Runtime Environment*: Choose Cloud Foundry.
   - *Space*: Select the appropriate space (depending on your company).
   - *Instance Name*: Enter an instance name.
4. In the *Parameters* step, select the `ESBMessaging.send` role or your custom role that you created and configured previously.
5. Choose *Next*.

Create the second instance for the SAP Cloud Integration OData API:

1. Access the sub-account used for SAP Cloud Integration.
2. Go to *Instances and Subscriptions*.
3. Create a new instance and add the following information in the *Basic Info* step:
   - *Service*: Process Integration (it-rt)
   - *Plan*: API (*Process Integration API access*)
   - *Runtime Environment*: Choose Cloud Foundry
   - *Space*: Select the appropriate space (depending on your company).
   - *Instance Name*: Enter an instance name.
4. Choose *Next*.
5. Select all of the following roles under *Parameters*:
   - `MessagePayloadsRead`
   - `AuthGroup_ReadOnly`
   - `AccessAllAccessPoliciesArtifacts` (If you use access policies.)

Creating a Service Key

After the instances have been created, to add the service key, do the following:

1. Select the row of the instance.
2. Go to the *Service Keys* tab.
3. Create the service key by entering its name.

For more information about creating service keys, see Creating Service Instance and Service Key for Inbound Authentication.

Using Access Policies

You can use access policies to restrict the integration artifacts – such as integration packages and iFlows – that are available in the *Data Integration Using SAP Cloud Integration* application job template in SAP IBP. This way you can ensure that only relevant iFlows are available for job scheduling.

If you use access policies, add the `AccessAllAccessPoliciesArtifacts` role to the *Parameters* of the instance that you created for the SAP Cloud Integration OData API.

For more information, see Managing Access Policies

In SAP IBP the communication scenario `SAP_COM_0B05` allows you to use both *OAuth 2.0 (Basic)* and *OAuth 2.0 (mTLS)* certificate-based authentication. We recommend that you use certificate-based authentication.

6 Creating Integration Flows Compatible With Application Jobs

If you want to create custom integration flows, you can learn here how scheduling information arrives from SAP IBP.

The configuration data from the application job arrives in a JSON format. The *SAP IBP Application Jobs – Schedule Sender* integration flow that's based on the `SchedulingMode` configuration parameter's value, routes the execution to the JMS or Data Store-based scheduling, calling either the *SAP IBP Application Jobs - Schedule with JMS* or *SAP IBP Application Jobs - Schedule without JMS* integration flows. They pass the scheduling information to the *SAP IBP Application Jobs – Schedule Receiver* integration flow via JMS or to the *SAP IBP Application Jobs – Schedule Receiver without JMS* integration flow via the Data Store adapter. In both cases, they call your custom integration flow via *ProcessDirect* passing the JSON data already transformed into a format acceptable for the scheduled integration flow.

Example of a JSON payload:

Sample Code

```
{
  "ProcessDirectId": "Example_ProcessDirectAddress",
  "External params": [
    {"ExampleExternalParam": "ExampleParamValue"},
    {"AnotherExternalizedParam": "AnotherValue"}
  ],
  "Additional params": [
    {"ParamA": "ValueA"},
    {"ParamB": "ValueB"}
  ]
}
```

The JSON payload is extracted, and the *ProcessDirectId* and every *Externalized Parameter* are set as header parameters.

Note

If you use the `DateFrom` and `DateTo` parameters, we evaluate possible xPath expressions when the data is extracted. Other parameters are passed to the called integration flow without evaluation and without any change in value.

If the value of the `DateFrom` and `DateTo` parameters can't be evaluated, because the xPath is invalid or uses any variable other than `IFlowStartTimestamp` for the functions, the integration flow throws an exception.

XPath expressions in other parameters are sent to the called integration flow without evaluation.

Simple *Expressions* (not the xPath expression, but expressions like ${header.parameter} or ${property.parameter}) that require value from the header parameter list or exchange property list are evaluated, if possible. If the referenced header parameter or exchange property can't be found, the value is passed to the called integration flow without any change.

The *Additional Parameters* are transformed to XML format and set as *Message body* without any changes to their values.

Sample Code

```
<AdditionalParams><ParamA>ValueA</ParamA><ParamB>ValueB</ParamB></
AdditionalParams>
```

Recommendations

If you want to schedule your integration flow from the SAP Cloud Integration *Application Jobs*, the *ProcessDirect Address* must be set as an externalized parameter. The name of the externalized parameter must be 'SAPIBP_PROCESS_DIRECT_ID', its value can be chosen freely.

Follow this process to set the *Address* as an *Externalized Parameter*:

1. Add the *Sender* adapter to your integration flow and connect it using *ProcessDirect* to your *Start Message*.
2. Configure the *ProcessDirect* on the *Connection* tab. Then choose *Externalize*.
3. Set parameter name to 'SAPIBP_PROCESS_DIRECT_ID' on the popup.

If the JMS capability is available, we recommend using the JMS-based scheduling. The Data Store-based approach has some disadvantages: it shares runtime node resources with other integration flows, and throughput is controlled by the number of available runtime nodes. There is no retry mechanism for failed jobs, consistent with the existing JMS-based behavior. While ordering follows the "oldest job first" principle, strict ordering guarantees are not provided. You can use the Data Store-based approach if you need application job scheduling capabilities without JMS.

If you already use JMS-based scheduling, you can continue without changes - the solution is fully backward compatible. However, if the *SAP IBP Application Jobs – Schedule Sender* integration flow is updated and deployed, you also need to deploy the *SAP IBP Application Jobs - Schedule with JMS* integration flow.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap