Box Adapter for SAP Integration Suite

Version 1.0.0 – November 2025

THE BEST RUN SAP

Contents

1 Introduction

1.1 Objective

This is the official guide for the Box Adapter for SAP Integration Suite. This guide covers all relevant information for integration developers to start working with the Microsoft Teams adapter. Read this guide carefully before using the Adapter.

1.2 Coding Samples

Any software coding and/or code lines/strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. The correctness and completeness of the Code given herein is not guaranteed.

1.3 Internet Hyperlinks

The documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. The availability and the correctness of this related information or the ability of this information to serve a particular purpose are not warranted.

1.4 Overview

Box is a platform that enables collaboration and management for various types of content.

Box adapter for SAP Integration Suite is a Cloud Integration capability available on SAP Integration Suite that allows you access and manage file and folder components available in the Box Application. The adapter works with the APIs of Box that allows developers to access and integrate its components into their applications enabling tasks like storage, access, and collaboration for documents and files.

The Box adapter has the following features:

- Enables access to Box for tasks like storage, access, and collaboration for documents and files.
- Supports secure authentication via **OAuth Client Credentials**.
- Supports **Basic** configuration type that provides a convenient processing capability for supported versions whereas **Advanced** enables proficient users to perform calls with greater control while connecting to any API endpoint.
- Supports **upload file** and **upload file version** via mutipart/form-data.

2 Installation and Configuration

This section details the file(s) available as part of the installation package and the prerequisites to configure the Box adapter.

The Box adapter is available as part of your SAP Integration Suite license.

2.1 Adapter Installation on Cloud Foundry

Before the adapter can be used in the Cloud Foundry environment, it must be deployed to the SAP Integration Suite tenant.

2.1.1 Prerequisites

To deploy the Box adapter, you must have access to the SAP Integration Suite license.

2.1.2 Procedure

You can deploy the adapter using the following methods:

The following installation procedure is compatible with Apache Camel 2, Apache Camel 3, and Edge Integration Cell (EIC) platform.

2.1.2.1 Adapter Installation by creating a New Integration Flow

The Box adapter is available for selection in the Sender and Receiver adapter list and can be deployed in the **Design** tab directly as you use it in an Integration flow.

Purpose

To install an adapter for use in your Integration flow.

Procedure

Go to **Design** workspace and select the integration package where you want to create a new Integration flow.

1. Click **Edit** to make the package editable.
2. Go to the **Artifacts** tab. Click **Add** and select **Integration Flow**.

3. Enter **Name** and **ID** for your flow. Additionally, select **Runtime Profile** from the drop-down and choose **Sender** and **Receiver** systems from the list . Finally, click **Add** to create the integration flow.
4. Go to the newly created integration flow and click **Edit** to make it editable.
5. In the integration flow, click **End** to add a **Connector** between the **End** and the **Receiver** Box.
   A drop-down with the available adapters appears. The Box adapter should show up in the list.
6. Select the Box Adapter from the list. The adapter is now imported which *triggers* an adapter deployment. Once the adapter is deployed, a success message is displayed.

   After the above steps are done, the adapter is successfully deployed in your Design workspace of the SAP Integration Suite tenant.

2.1.2.2 Adapter Installation without Creating a New Integration Flow

The following procedure explains how the adapter is migrated from the Discover workspace to the Design workspace of the SAP Integration tenant.

This method is useful for scenarios where integration flow packages are migrated from development to a higher environment such as Production.

The adapter can be imported into the Design workspace without creating an integration flow. Use the Transport Management Service (TMS) to import/transport the adapter to a higher environment. Alternatively, if the TMS is not available in the landscape, the adapter package can be imported to the Design workspace by copying it from the Discover workspace.

Purpose

To copy the integration package from the Discover workspace and import the Box adapter to the Design workspace, follow these steps:

Procedure

1. Go to **Discover** workspace.
2. In the search box, search for **Box Adapter for SAP Integration Suite** package.
3. Select the package and click **Copy**. This copies the package from the Discover workspace to the Design workspace.
4. Go to Design workspace and select the copied **Box Adapter for SAP Integration Suite** package.
5. In the **Actions** tab of the selected package, click **Deploy**. This completes the adapter deployment to the Design workspace.

2.2 Monitor the Deployment Status

After the adapter deployment is complete, you can check the status in the **Monitor** section.

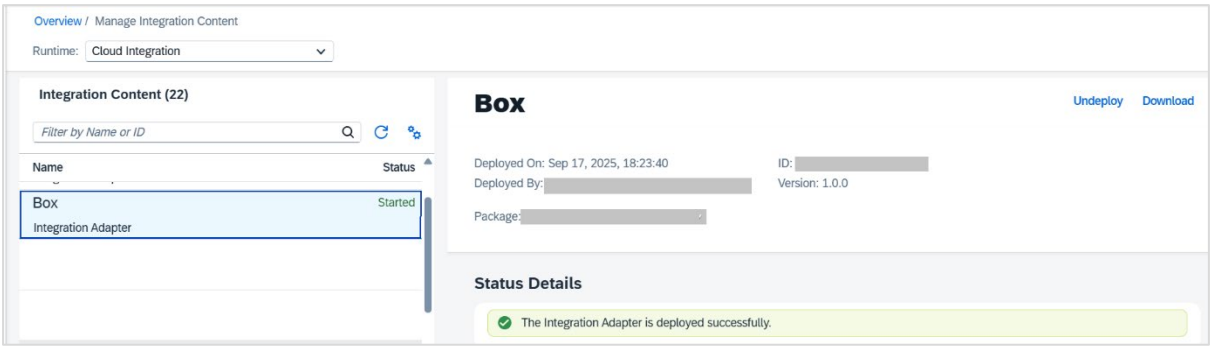
Purpose

To check the status of the deployed adapter.

Procedure

1. Under the **Monitor** tab, click **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Integration Content** section and click **All**. This opens the **Integration Content** page with a list of all the deployed adapters.

Here, you can check and confirm the deployment status of your adapter.



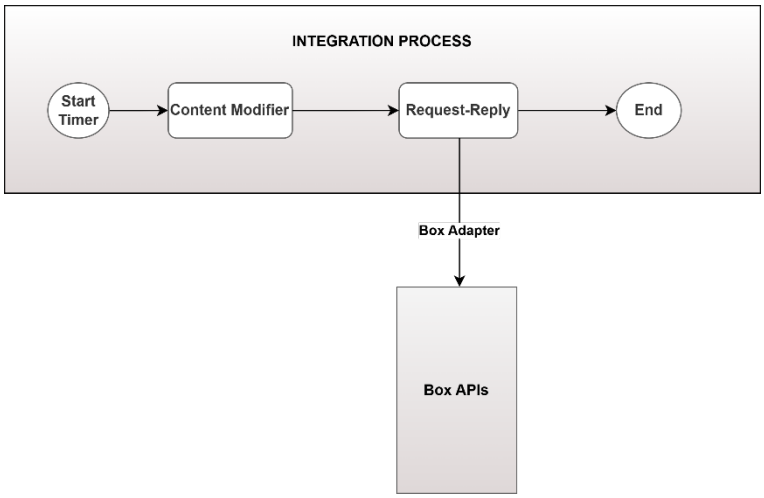
3 Getting Started: Box Adapter

This section explains how to configure the Box adapter for SAP Integration Suite. You can find information about adapter architecture, application configuration, and authentication for Box Adapter.

3.1 Architecture Overview

How Box Receiver Adapter Works: The Box Receiver adapter (as the name suggests) is designed to function as a receiver adapter. In such a scenario where the adapter is used as a receiver adapter, SAP Integration Suite acts as the initiator of the calls.

SAP Integration Suite tenant sends the operation request to Box (this is a receiver system), Box Receiver Adapter works on the request and sends the data back to the SAP Integration Suite tenant.



3.2 Application Configuration

The links below contain basic information to help you get started with the Box Application.

- To get started with Box Application, see Box Overview.
- For detailed information about Box APIs, see Box API Reference.
- You must create a new Box application on the Developers console. For more information, see Create a Box Platform App.

> You must have access to the developer's console in Box.

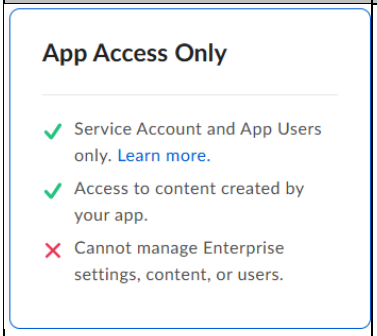
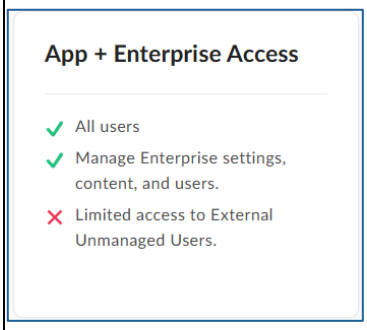
- To authenticate your application's identity using a client ID and client secret, see Create Client Credentials in Box.

> Ensure you have correctly configured and set up authentication and app level access before calling the Box application via the Box Adapter in an integration scenario.

- The application access level determines which users and content your app may access.

    a) **App Access Only:** It provides access for Service Account and App Users.
    b) **App + Enterprise Access:** It has App level access along with access for any existing managed user in the enterprise.

For information about access levels for your Box Application, see Application Access. For more information, see the table below.

Scenarios	Description
	• If **Box_subject_type** is set to `enterprise`, token generation is allowed, and all operations are allowed via service account. • If **Box_subject_type** is set to `user`, token generation is not allowed with managed user id.
	• If **Box_subject_type** is set to `enterprise`, token generation is allowed, and all operations are allowed via service account. • If **Box_subject_type** is set to `user`, token generation is allowed for all managed users. But users can perform CRUD(Create, Read, Update & Delete) operations in shared files/folders only if added as collaborator and not at an App Access level.

3.3 Authentication

This section details the authentication mechanism supported by the Box Adapter in the SAP Integration Suite.

The Box adapter supports **Client Credentials.** You can securely store the security artifacts in SAP Security Material. This ensures that credentials can be safely provided to the Adapter.

Before setting up the authentication, ensure you have finished the required steps as directed in Application Configuration.

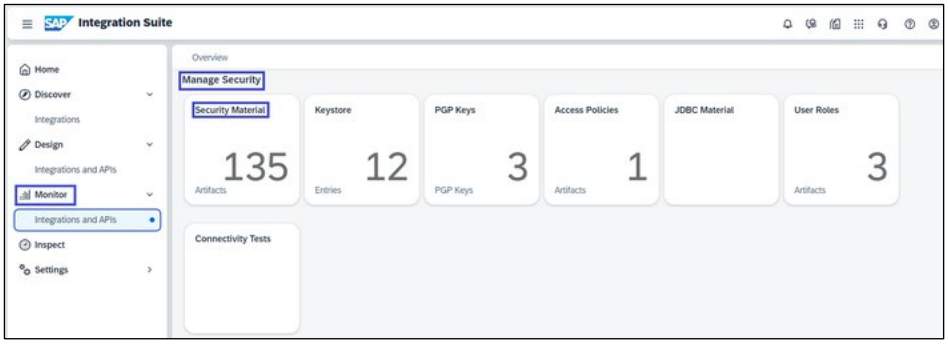
3.3.1 Creating OAuth2 Client Credentials artifact

Purpose

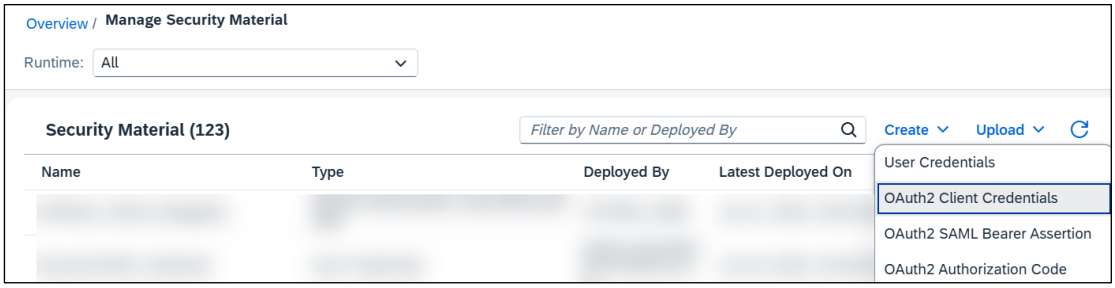
To create credentials in Security Material for **OAuth2 Client Credentials**.

Procedure

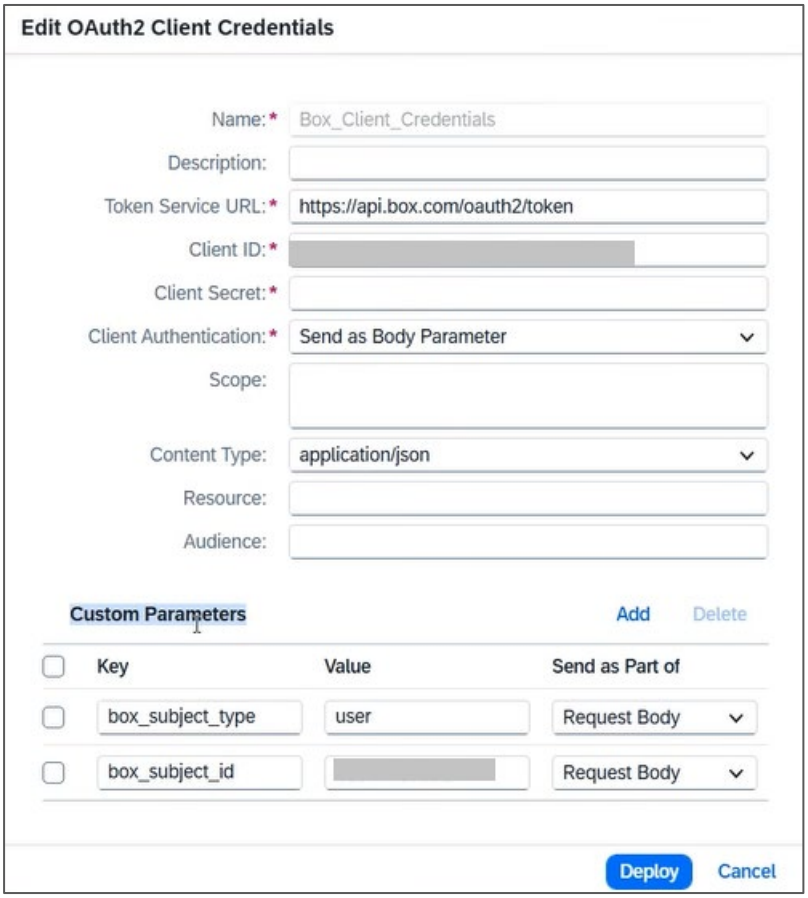
1. In SAP Integration Suite, navigate to **Monitor** > **Integrations and APIs**. This opens the **Overview** page.
2. On the **Overview** page, go to **Manage Security** section and click **Security Material**.



3. On **Manage Security Material** page, click **Create** to select **OAuth2 Client Credentials** from the dropdown.



4. In the **Create OAuth2 Client Credentials** popup, provide the details below.



Parameter	Description
Name	Enter a name for the credentials. The integration flow configuration uses the name you enter.
Description	Enter an optional description.
Token Service URL	`https://api.box.com/oauth2/token`
Client ID	Go to the **Box Console** screen and copy and paste the appropriate value for client ID.
Client Secret	Go to the **Box Console** screen and copy and paste the appropriate value for client Secret.
Client Authentication	Select **Send as Body Parameter** from the dropdown list
Content Type	Select **application/json** from the dropdown list.

5. Click **Deploy** to complete the process.

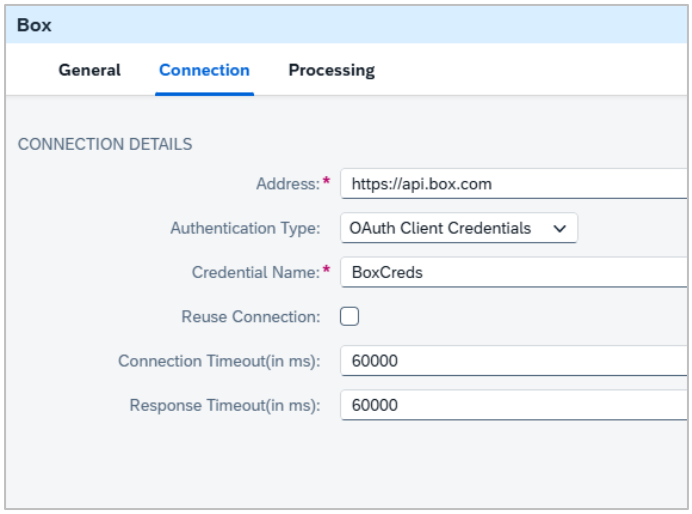
When you refresh the **Manage Security Material** page, the new artifact is displayed (with Type **OAuth2 Client Credentials**) in the artifact table.

4 Box Adapter Configuration

This section describes the parameters to be configured for your Box adapter. You need to configure the **Connection** and **Processing** tabs. A description and example usage for every field has been added.

4.1 Connection

The Connection tab contains connection and authentication parameters for Box. The Security artifact created in Authentication, should be used in the **Connection tab** of the Adapter as shown in the sample figure below.



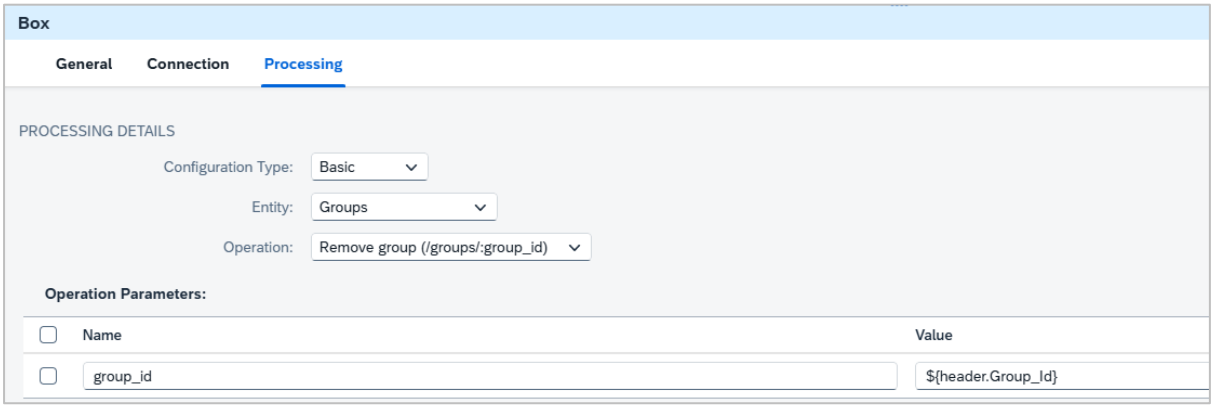
The table below contains the complete list of fields that can be found in the **Connection** tab.

Parameter	Description
Address	Specify the address to connect to Box. Ensure that you specify the address corresponding to the operation as different operations might have different addresses. Example: • `https://api.box.com` • `https://upload.box.com`
Authentication Type	Select the required authentication type to be used. Currently only **OAuth Client Credentials** is supported.

Parameter	Description
Credential Name	Specify the name of OAuth2 Client Credentials artifact that stores the client id and client secret (Client Credentials Grant). For more information, see Authentication.
Reuse Connection	Enable to reuse the connection.
Connection Timeout(in ms)	Specify the maximum waiting time (in milliseconds) for the connection to be established. Example: `60000`
Response Timeout(in ms)	Specify the maximum waiting time (in milliseconds) for a response message. Example: `60000`

4.2 Processing

The **Processing** tab allows you to configure the operations that can be performed using Box Adapter. The below image shows a sample operation portraying **Remove group** operation.



The table below contains the complete list of fields that can be found in the **Processing** tab.

Parameter	Description
Configuration Type	Select the configuration type: • **Basic** to use the dropdowns and parameter text fields • **Advanced** to provide the relative URL.

Parameter	Description
Entity	Select the entity based on which the operation will be performed.
Operation	Select the operation to be performed.
Query Parameter	Specify the name of the file. The file name must be unique within its parent folder. Example: `readme.txt`
Parent ID	Specify the ID of the parent folder to upload the file to it. Note: Use `0` to specify user's root folder.
Operation Parameters	Specify the operation parameters as a key value pair.
Name	Specify the key for the operation parameter. Example: `file_id`
Value	Specify the value for the operation parameter. Example: `131231233`
HTTP Method	Select the required HTTP method from the available dropdown: • **DELETE** • **GET** • **PATCH** • **POST** • **PUT**
Relative URL	Specify the endpoint path, excluding the Host. Example: `/2.0/search`
Query	Specify the query that should be transferred with the HTTP request. Example: `param1=value1&param2=value2`
Request Headers	Specify a list of custom headers, separated by a pipe (|), to be sent to the target system. Use an asterisk (*) to send all custom headers to the target system. All Camel-specific headers and HTTP protocol headers except "date" are excluded by default even if you specify them.

Parameter	Description
Response Headers	Specify a list of headers, separated by a pipe (|), coming from the target system's response to be received in the message. Use an asterisk (*) to receive all the headers from the target system, which is also the default value. All Camel-specific headers and HTTP protocol headers except "date" are excluded by default even if you specify them.

5 Box Operations

This section lists and describes some of the operations supported by the Box Adapter.

For detailed information about API calls used by the below operations, see Box API Reference.

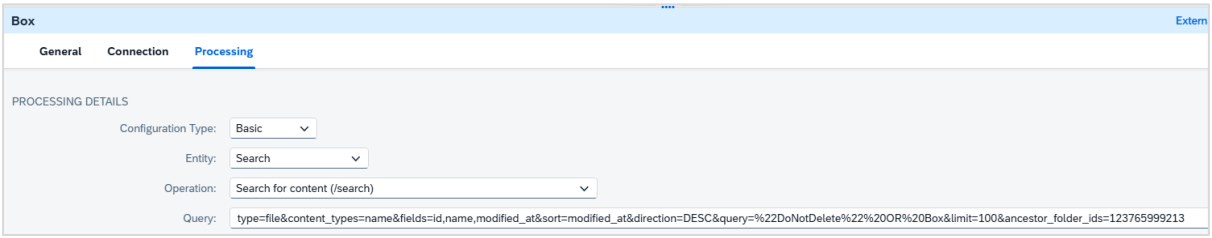
When the `fields` parameter is used to query additional information about the items, only those fields and a few base fields (`id`, `type`, `name` etc.) are returned. Any fields that were originally in the response would now have to be requested explicitly.

5.1 Search for content

Search for files, folders, web links, and shared files based on the query and returns the content for the matching object (files, folders).

You can perform the operation in two ways:

a) If **Configuration Type** is set to **Basic**:



The following example searches for files (`type=file)` in a folder (`ancestor_folder_ids`) by filenames (`content_types=name`) matching the keyword "`Box`" or "`DoNotDelete`".

You can also sort Box search results (`sort=field1&direction=DESC`) and specify the fields to be returned (`fields=field1,field2,field3`).

For more information regarding the complete set of fields for the adapter, see Processing.

Parameter	Value
Configuration Type	**Basic**
Entity	**Search**
Operation	**Search for content (/search)**

Query	`type=file&content_types=name&fields=id,name,modified_at&sort=modified_at&direction=DESC&query=%22DoNotDelete%22%20OR%20Box&limit=100&ancestor_folder_ids=<enter_folder_id>`
	
	Note:
	
	• Box supports search operators. For more information about the list of supported operators, see Query Operators.
	• While using space( ) and double quotes (""), ensure that you encode the query.
	
	Example:
	
	`query="DoNotDelete" OR Box`
	
	must be encoded as
	
	`query=%22DoNotDelete%22%20OR%20Box`

In some cases, an index might not be updated even after 10 minutes. For more information about Search operation, see Box Search.

b) If **Configuration Type** is set to **Advanced**

Parameter	Value
Configuration Type	**Advanced**
HTTP Method	**GET**
Relative URL	`/2.0/search`
Query	`type=file&content_types=name&fields=id,name,modified_at&sort=modified_at&direction=DESC&query=%22DoNotDelete%22%20OR%20Box&limit=100&ancestor_folder_ids=<enter_folder_id>`

5.2 Upload File

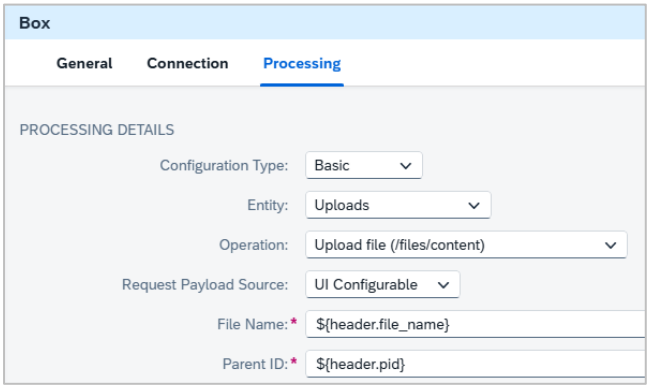
To upload a file, you must provide the content of the file, a file name, and the parent folder ID.

> While uploading large files to Box, ensure that you adjust your adapter configurations. Increase values for **Connection Timeout** and **Response Timeout**. As a best practice upload files more than 50MB using upload chunks via **Advanced** configuration.

The following example demonstrates uploading a file to Box.

You can perform the operation in two ways:

a) **Request Payload Source** is set to **UI Configurable**



For more information regarding the complete set of fields for the adapter, see Processing.

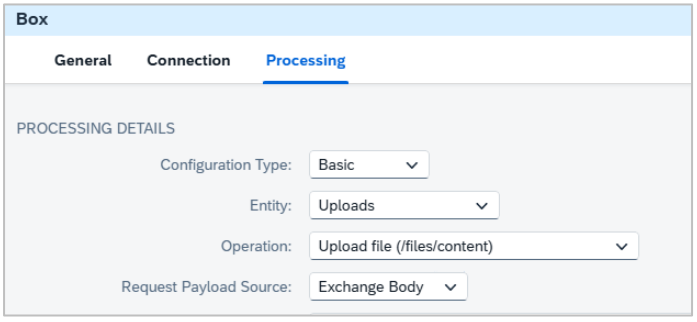
Parameter	Value
Configuration Type	**Basic**
Entity	**Uploads**

Parameter	Value
Operation	**Upload file version (/files/:file_id/content)**
Request Payload Source	**UI Configurable**
File Name	`file.txt`
Parent ID	Specify parent id. **Note:** You can find this value in the address bar link when viewing the file in the Box Explorer on a web browser. Example: `612055232281`

Payload for file content:

```
This is the file content being uploaded to Box.
```

b) **Request Payload Source** is set to **Exchange Body**



Along with the payload, ensure that you provide the specific content type header that defines the request payload structure.

Header:

```
Content-Type "multipart/form-data; boundary=------------------------
----boundary123"
```

Payload:

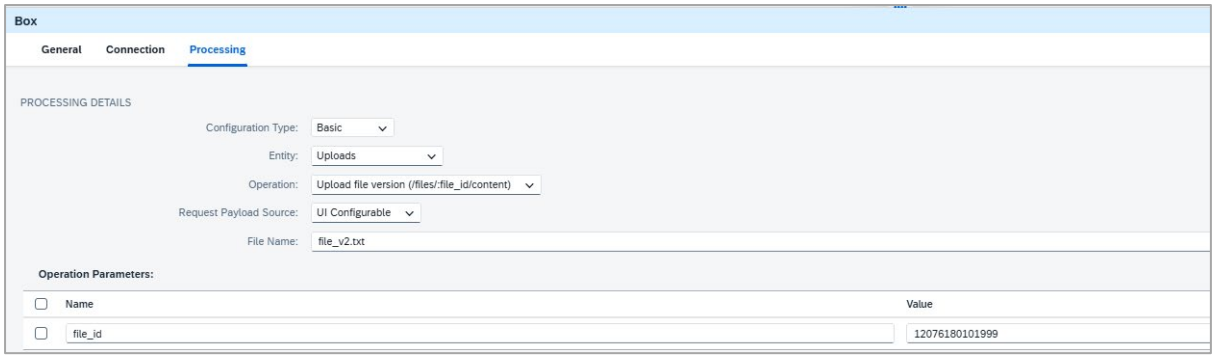
```
----------------------------boundary123
Content-Disposition: form-data; name="attributes"
Content-Type: application/json

{
  "name": "example.txt",
  "parent": { "id": "123456789" }
}
----------------------------boundary123
Content-Disposition: form-data; name="file";
  filename="example_file_name.txt"
Content-Type: text/plain

Hello, this is the file content being uploaded to Box.
----------------------------boundary123--
```

5.3 Upload file version

- Ensure the Connection tab has Address set to `https://upload.box.com/`
- The attributes part of the body must come before the file part. Requests that do not follow this format when uploading the file will receive a HTTP 400 error with a `metadata_after_file_contents` error code.

In the below example, you must provide the file in binary format.

Sample Text for Payload: `File contains security related information.`

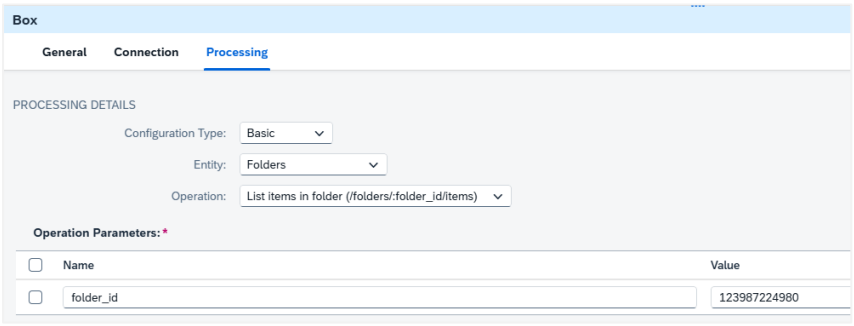


For more information regarding the complete set of fields for the adapter, see Processing.

Parameter	Value
Configuration Type	**Basic**

Parameter	Value
Entity	**Uploads**
Operation	**Upload file version (/files/:file_id/content)**
Request Payload Source	**UI Configurable**
File Name	`Syntax: file_name.extension` Ensure you provide the proper file name including extension, or you can leave it blank as it is automatically renamed for a new version.
Operation Parameters	
Name	Specify key for operation parameter. `file_id`
Value	Specify value for file id. Example: `12076180101999` **Note:** You can find this in the address bar link when viewing the file in the Box Explorer on a web browser.

5.4 List Folder Items



For more information regarding the complete set of fields for the adapter, see Processing.

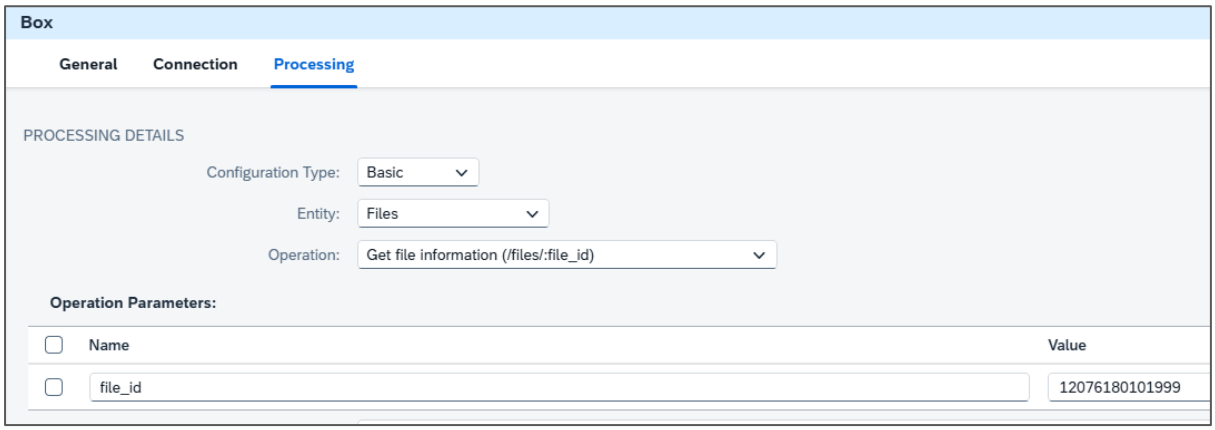
Parameter	Value
Configuration Type	**Basic**

Parameter	Value
Entity	**Folders**
Operation	**List items in folder (/folders/:folder_id/items)**
Operation Parameters	
Name	Specify key for operation parameter. `folder_id`
Value	Specify value for folder id. Example: `12076180101999` **Note:** You can find this in the address bar link when viewing the folder in the Box Explorer on a web browser.

5.5 Download File

The following scenario demonstrates downloading a file from Box by specifying its `file_id`.

This operation returns content of the file in binary format.



For more information regarding the complete set of fields for the adapter, see Processing.

Parameter	Value
Configuration Type	**Basic**
Entity	**Downloads**
Operation	**Download file (/files/:file_id/content)**

Parameter	Value
Operation Parameters	
Name	Specify key for operation parameter. `file_id`
Value	Specify value for file id. Example: `12076180101999` **Note:** You can find this in the address bar link when viewing the file in the Box Explorer on a web browser.

For more information regarding the complete set of fields, see Processing.

Box operations introduced in 2025 support the `box-version` request header. The Box adapter supports these operations via the Advanced configuration.For more information, see API Versioning Strategy.

You can configure headers as shown below: