

# Table of contents

---

- [Table of contents](#)
- [JSON-Create](#)
  - [Description](#)
  - [Compatibility](#)
  - [Getting Started](#)
  - [Output Connections](#)
  - [Flow Element](#)
    - [General Properties](#)
    - [Working Mode Depending Properties](#)
      - [Multi-line](#)
      - [XML](#)
      - [XML-Dataset](#)
      - [External XML-File](#)
  - [Examples](#)
    - [Multi-line to JSON](#)
    - [XML to JSON](#)
  - [Error handling](#)
    - [Private data](#)

# JSON-Create

---

## Description

---

With JSON-Create you can generate JSON files from a multiline field or from an XML file. The XML conversion comes with several expert options to satisfy all your needs.

## Compatibility

---

Switch Fall 2022 and higher.

## Getting Started

---

Use one of our sample flows and drop a sample file into the flow.

## Output Connections

---

This app requires one incoming connection - more incoming connections are allowed. The app supports traffic light outgoing connections of the following types:

- Log success: carries the created job if `Append Result as Dataset` is set to `false`
- Data error: carries the incoming job if the operation fails at the first attempt.
- Data success: carries the incoming job after the operation succeeds. If there are no data success connections the output is simply suppressed (with logging a warning).

# Flow Element

---

## General Properties

Property	Value	Description
<i>Generate JSON From</i>	enum [ Multi-line   XML-File   XML-Dataset   External XML-File ]	Defines the source of the input file
<i>Append Result As Dataset</i>	Boolean	Defines a JSON (optionally with Switch variables)
<i>Job Name</i>	String	Defines the name proper of the created JSON
<i>Dataset Name</i>	String	Defines the dataset name of the JSON

## Working Mode Depending Properties

### Multi-line

Property	Value	Description
<i>JSON</i>	String[]	Defines a JSON (optionally with Switch variables)

### XML

Property	Value	Description
<i>Expert Settings</i>	Boolean	Enable or disable additional settings for the conversion from XML to JSON - click <a href="#">HERE</a> further details

Defaults:

```

{
  preserveOrder: false,
  attributeNamePrefix: "",
  attributesGroupName: "$",
  textNodeName: "_",
  ignoreAttributes: false,
  removeNSPrefix: false,
  allowBooleanAttributes: false,
  parseTagValue: false,
  parseAttributeValue: false,
  trimValues: true,
  cdataPropName: false,
  stopNodes: [],
  alwaysCreateTextNode: false,
  commentPropName: false,
  unpairedTags: [],
  processEntities: true,
  htmlEntities: false,
  ignoreDeclaration: false,
  ignorePiTags: true
}

```

### **XML-Dataset**

Property	Value	Description
<i>Dataset Name</i>	String	Name of the XML-Dataset that should be converted to JSON

### **External XML-File**

Property	Value	Description
<i>File Path</i>	String	Path to external XML file
<i>Delete after Injection</i>	Boolean	Defines if the external file should be deleted or not after it is injected as Switch job



# XML to JSON

## Input

```
<?xml version="1.0" encoding="UTF-8"?>
<orders version="1.0">
  <order>
    <ID>1</ID>
    <name>Order1</name>
  </order>
  <order>
    <ID>2</ID>
    <name>Order1</name>
  </order>
</orders>
```

## Result

```
{
  "orders": {
    "order": [
      {
        "ID": "1",
        "name": "Order1"
      },
      {
        "ID": "2",
        "name": "Order1"
      }
    ],
    "$": {
      "version": "1.0"
    }
  }
}
```

## Error handling

---

This app uses two types of errors:

- job data: if an handled error occurs (e.g. wrong file format), the error message is logged in the switch messages.
- job fail: if any other error occurs, job will fail and gets sent to the problem jobs folder. The thrown error gets logged as error and can be looked up in the switch messages.

### Private data

The following private data tags will be set if an error occurs:

Tag	Value   Type	Description
lastErrorElement	String	the name of the flow element
lastErrorId	jsonCreateError	
lastErrorCode	Number	an error code that defines the type of error that occurred
lastErrorMessage	String	detailed error message

### Error Codes:

```
enum ERROR_CODES {
  generalError = 0,
  fileHandlingError = 1,
  fileFormatError = 2,
  conversionError = 3,
  invalidParameterValue = 4,
  parsingError = 5,
}
```