Base64

Description

Base64 allows you to encode/decode data to/from base64.

Base64 encoding schemes are commonly used when there is a need to encode binary data, especially when that data needs to be stored and transferred over media that are designed to deal with text. This encoding helps to ensure that the data remains intact without modification during transport. Base64 is used commonly in a number of applications including email via MIME, as well as storing complex data in XML or JSON.

Compatibility

Switch 2020 Fall or higher.

Connections

Base64 has traffic light outgoing connections for success and for error.
Error log contains an error message. Success log contains the processed data (if asked).

Use cases

- Communicate with some web services
- Embed an image in an HTML without hosting it
- Decode a file embedded in a XML or a HTTP response

Flow element properties

- **Mode**

  Where the data to process comes from.

  If you plan to process large data, Switch may not always be able to retrieve the string value within an XML dataset.
  In this case, you can choose "Select string in XML" and specify the xpath to the value you want in an input XML file or an existing XML dataset. The value will be selected within the app.

  Possible values:

    - File
    - String
    - Select string in XML

  - String to process
    String to be encoded or decoded.

  - Select from
    Select the value from an input XML file or an existing XML dataset.

    Possible values:

    - Input XML file
    - XML dataset

    - Dataset name
      Specify the dataset name.

  - XPath
    Write the xpath to the string you want to process in the XML file or dataset.

  - Encoding
    Encoding used in the file to process. Choose “binary” if the input file is a PDF, an image, or any non-text file.

- **Action**

  Whether to encode or decode the input data.

Possible values:

- Encode
- Decode

- **Write to private data**

  If the result should be written in a private data key.

  You should use this property if the result is text based.

  Note that there is a limitation to the size of the private data you can write. So you better not use it to process large data.

  - Private data
    Private data key where the result will be written.

- **Write to log**

  If the result should be written in the output success log.
  You should use this property if the result is text based.

- **Write to file**

  If the result should be written in a file.

  - File name
    Name of the file where the result will be written.

  - Content
    Choose “Result” if you want to write the processed data to the file without any change.

    - Output content
      Specify the content you want.

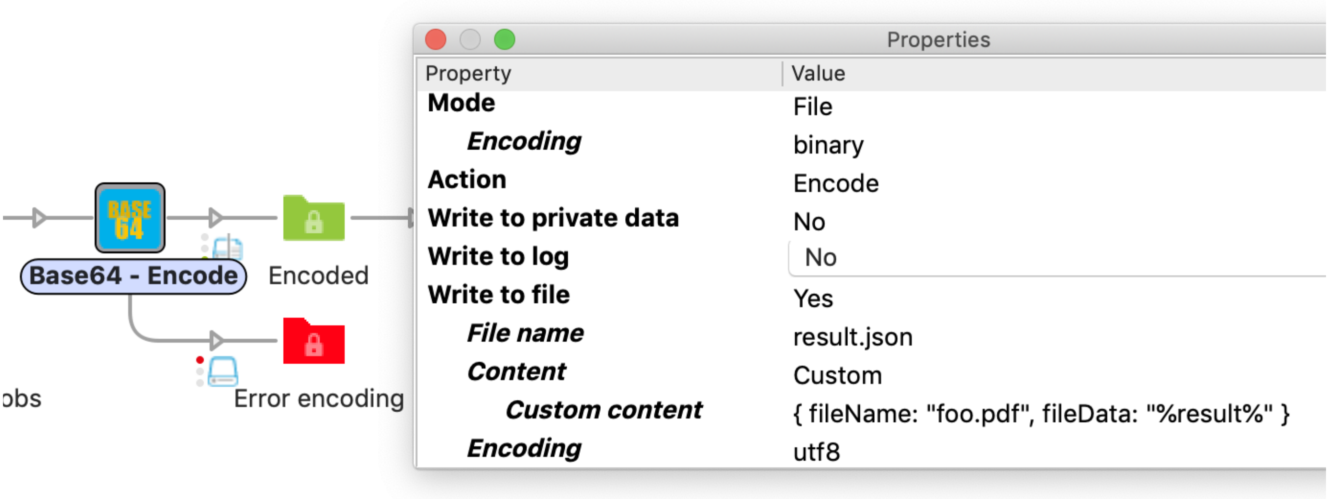
      Use “%result%” as a substitute to refer to the processed data.

      This property can be useful if the output is a text based file.

  - Encoding
    Encoding used in the written file. Choose “binary” to write a PDF, an image, or any non-text file.

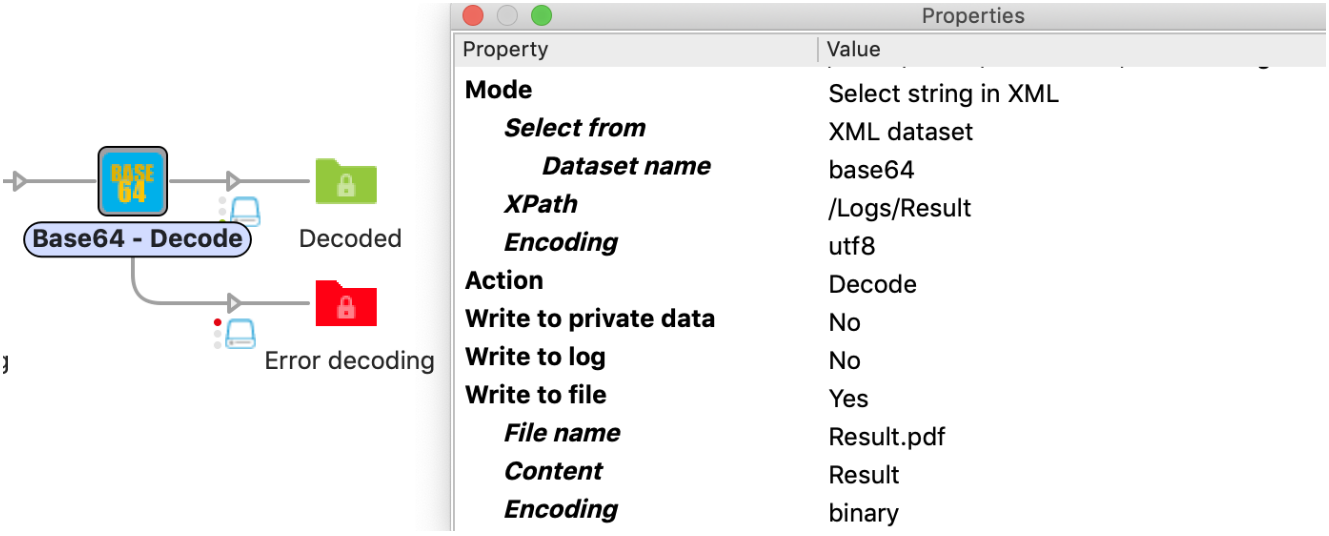
Example flows

Encode



Property	Value
Mode	File
Encoding	binary
Action	Encode
Write to private data	No
Write to log	No
Write to file	Yes
File name	result.json
Content	Custom
Custom content	{ fileName: "foo.pdf", fileData: "%result%" }
Encoding	utf8

Decode



Property	Value
Mode	Select string in XML
Select from	XML dataset
Dataset name	base64
XPath	/Logs/Result
Encoding	utf8
Action	Decode
Write to private data	No
Write to log	No
Write to file	Yes
File name	Result.pdf
Content	Result
Encoding	binary