**Using the Attachment Processor in a Pipeline**

**Table 1. Attachment options**

| **Name** | **Required** | **Default** | **Description** |
| --- | --- | --- | --- |
| field | yes | - | The field to get the base64 encoded field from |
| target\_field | no | attachment | The field that will hold the attachment information |
| indexed\_chars | no | 100000 | The number of chars being used for extraction to prevent huge fields. Use -1 for no limit. |
| indexed\_chars\_field | no | null | Field name from which you can overwrite the number of chars being used for extraction. See indexed\_chars. |
| properties | no | all properties | Array of properties to select to be stored. Can be content, title, name, author, keywords, date, content\_type, content\_length, language |
| ignore\_missing | no | false | If true and field does not exist, the processor quietly exits without modifying the document |

For example, this:

PUT \_ingest/pipeline/attachment

{

"description" : "Extract attachment information",

"processors" : [

{

"attachment" : {

"field" : "data"

}

}

]

}

PUT my\_index/\_doc/my\_id?pipeline=attachment

{

"data": "e1xydGYxXGFuc2kNCkxvcmVtIGlwc3VtIGRvbG9yIHNpdCBhbWV0DQpccGFyIH0="

}

GET my\_index/\_doc/my\_id

Copy as

Returns this:

{

"found": true,

"\_index": "my\_index",

"\_type": "\_doc",

"\_id": "my\_id",

"\_version": 1,

"\_seq\_no": 22,

"\_primary\_term": 1,

"\_source": {

"data": "e1xydGYxXGFuc2kNCkxvcmVtIGlwc3VtIGRvbG9yIHNpdCBhbWV0DQpccGFyIH0=",

"attachment": {

"content\_type": "application/rtf",

"language": "ro",

"content": "Lorem ipsum dolor sit amet",

"content\_length": 28

}

}

}

To specify only some fields to be extracted:

PUT \_ingest/pipeline/attachment

{

"description" : "Extract attachment information",

"processors" : [

{

"attachment" : {

"field" : "data",

"properties": [ "content", "title" ]

}

}

]

}

Copy as cURL[View in Console](http://localhost:5601/app/kibana#/dev_tools/console?load_from=https://www.elastic.co/guide/en/elasticsearch/plugins/6.8/snippets/24.console)

Extracting contents from binary data is a resource intensive operation and consumes a lot of resources. It is highly recommended to run pipelines using this processor in a dedicated ingest node.

Limit the number of extracted chars

To prevent extracting too many chars and overload the node memory, the number of chars being used for extraction is limited by default to 100000. You can change this value by setting indexed\_chars. Use -1 for no limit but ensure when setting this that your node will have enough HEAP to extract the content of very big documents.

You can also define this limit per document by extracting from a given field the limit to set. If the document has that field, it will overwrite the indexed\_chars setting. To set this field, define the indexed\_chars\_field setting.

For example:

PUT \_ingest/pipeline/attachment

{

"description" : "Extract attachment information",

"processors" : [

{

"attachment" : {

"field" : "data",

"indexed\_chars" : 11,

"indexed\_chars\_field" : "max\_size"

}

}

]

}

PUT my\_index/\_doc/my\_id?pipeline=attachment

{

"data": "e1xydGYxXGFuc2kNCkxvcmVtIGlwc3VtIGRvbG9yIHNpdCBhbWV0DQpccGFyIH0="

}

GET my\_index/\_doc/my\_id

Copy as cURL[View in Console](http://localhost:5601/app/kibana#/dev_tools/console?load_from=https://www.elastic.co/guide/en/elasticsearch/plugins/6.8/snippets/25.console)

Returns this:

{

"found": true,

"\_index": "my\_index",

"\_type": "\_doc",

"\_id": "my\_id",

"\_version": 1,

"\_seq\_no": 35,

"\_primary\_term": 1,

"\_source": {

"data": "e1xydGYxXGFuc2kNCkxvcmVtIGlwc3VtIGRvbG9yIHNpdCBhbWV0DQpccGFyIH0=",

"attachment": {

"content\_type": "application/rtf",

"language": "sl",

"content": "Lorem ipsum",

"content\_length": 11

}

}

}

PUT \_ingest/pipeline/attachment

{

"description" : "Extract attachment information",

"processors" : [

{

"attachment" : {

"field" : "data",

"indexed\_chars" : 11,

"indexed\_chars\_field" : "max\_size"

}

}

]

}

PUT my\_index/\_doc/my\_id\_2?pipeline=attachment

{

"data": "e1xydGYxXGFuc2kNCkxvcmVtIGlwc3VtIGRvbG9yIHNpdCBhbWV0DQpccGFyIH0=",

"max\_size": 5

}

GET my\_index/\_doc/my\_id\_2

Copy as cURL[View in Console](http://localhost:5601/app/kibana#/dev_tools/console?load_from=https://www.elastic.co/guide/en/elasticsearch/plugins/6.8/snippets/26.console)

Returns this:

{

"found": true,

"\_index": "my\_index",

"\_type": "\_doc",

"\_id": "my\_id\_2",

"\_version": 1,

"\_seq\_no": 40,

"\_primary\_term": 1,

"\_source": {

"data": "e1xydGYxXGFuc2kNCkxvcmVtIGlwc3VtIGRvbG9yIHNpdCBhbWV0DQpccGFyIH0=",

"max\_size": 5,

"attachment": {

"content\_type": "application/rtf",

"language": "ro",

"content": "Lorem",

"content\_length": 5

}

}

}

Using the Attachment Processor with arrays[edit](https://github.com/elastic/elasticsearch/edit/6.8/docs/plugins/ingest-attachment.asciidoc)

To use the attachment processor within an array of attachments the [foreach processor](https://www.elastic.co/guide/en/elasticsearch/reference/6.8/foreach-processor.html) is required. This enables the attachment processor to be run on the individual elements of the array.

For example, given the following source:

{

"attachments" : [

{

"filename" : "ipsum.txt",

"data" : "dGhpcyBpcwpqdXN0IHNvbWUgdGV4dAo="

},

{

"filename" : "test.txt",

"data" : "VGhpcyBpcyBhIHRlc3QK"

}

]

}

In this case, we want to process the data field in each element of the attachments field and insert the properties into the document so the following foreach processor is used:

PUT \_ingest/pipeline/attachment

{

"description" : "Extract attachment information from arrays",

"processors" : [

{

"foreach": {

"field": "attachments",

"processor": {

"attachment": {

"target\_field": "\_ingest.\_value.attachment",

"field": "\_ingest.\_value.data"

}

}

}

}

]

}

PUT my\_index/\_doc/my\_id?pipeline=attachment

{

"attachments" : [

{

"filename" : "ipsum.txt",

"data" : "dGhpcyBpcwpqdXN0IHNvbWUgdGV4dAo="

},

{

"filename" : "test.txt",

"data" : "VGhpcyBpcyBhIHRlc3QK"

}

]

}

GET my\_index/\_doc/my\_id

Copy as cURL[View in Console](http://localhost:5601/app/kibana#/dev_tools/console?load_from=https://www.elastic.co/guide/en/elasticsearch/plugins/6.8/snippets/27.console)

Returns this:

{

"\_index" : "my\_index",

"\_type" : "\_doc",

"\_id" : "my\_id",

"\_version" : 1,

"\_seq\_no" : 50,

"\_primary\_term" : 1,

"found" : true,

"\_source" : {

"attachments" : [

{

"filename" : "ipsum.txt",

"data" : "dGhpcyBpcwpqdXN0IHNvbWUgdGV4dAo=",

"attachment" : {

"content\_type" : "text/plain; charset=ISO-8859-1",

"language" : "en",

"content" : "this is\njust some text",

"content\_length" : 24

}

},

{

"filename" : "test.txt",

"data" : "VGhpcyBpcyBhIHRlc3QK",

"attachment" : {

"content\_type" : "text/plain; charset=ISO-8859-1",

"language" : "en",

"content" : "This is a test",

"content\_length" : 16

}

}

]

}

}

Note that the target\_field needs to be set, otherwise the default value is used which is a top level field attachment. The properties on this top level field will contain the value of the first attachment only. However, by specifying the target\_field on to a value on \_ingest.\_value it will correctly associate the properties with the correct attachment.

**1 more Example from Web**

<https://github.com/rahulsinghai/elasticsearch-ingest-attachment-plugin-example>