mantis

# Batch Processor Documentation

## Publisher Integration

**Prepared By:** Callum Mcadam

**Version:** 1.1

**Last Updated:** 3/3/2025 (DD/MM/YYYY)

**Disclaimer:** This document is actively maintained and will be updated regularly to reflect improvements in the system. While the core process remains consistent, refinements may be made to enhance performance and functionality. As a result, this document is subject to change.

## Table of Contents

# 1. Overview of the Mantis Batch Processor

The Mantis Batch Processor is a batch processing service designed to evaluate and return brand safety ratings for large sets of URLs asynchronously. Leveraging Amazon S3 for data storage, the service processes URLs in bulk.

The typical workflow involves uploading an hourly batch file containing a list of URLs to the input bucket. The Mantis Batch Processor analyses the URLs and returns the brand safety ratings as JSON files, which are stored in the output bucket. These JSON files are then retrieved and integrated into the publisher's CMS to ensure that brand safety information is available for content management and advertising decisions.

# 2. Authentication and Credentials

To securely access the Mantis Batch Processor, you will need specific credentials: You will receive your access credentials from your account manager.

**Access Credentials for the Bucket:** These are unique identifiers assigned to your account for accessing the Amazon S3 bucket. You must use these credentials to authenticate your requests to the S3 bucket for uploading and retrieving data files.

**Customer ID:** This identifier distinguishes your specific instance of the Mantis Batch Processor service. It helps ensure that your data submissions are routed correctly within our systems. Like the Access Credentials, your Customer ID will be provided by your Account Manager.

# 3. Connecting to Amazon S3

## 3.1. Direct Bucket Access

Command Line Integration: You can connect to your Amazon S3 buckets using several command line tools:

- AWS CLI: Ideal for managing S3 resources directly from your terminal.
- s3fs: A utility that allows you to mount S3 buckets as local file system directories.
- Rclone: A powerful command line program to manage files on cloud storage.

SDK Integration: For application-based integrations, you can use one of the following SDKs:

- Python: Utilise the boto3 library for comprehensive AWS S3 management.
- Node.js: Use the AWS SDK for JavaScript to interact with S3.
- Java: Implement the AWS SDK forJava to access S3 resources.
- Go: Use the AWS SDK for Go to manipulate S3 data programmatically.

Desktop Applications:

- CyberDuck: A user-friendly desktop application that supports S3 connections.

## 3.2. Pre-signed URLs (Coming Soon)

In addition to direct buc ket access, we will soon offer the ability to use pre-signed URLs. Pre signed URLs will allow you to grant temporary access to your S3 resources without sharing your actual access credentials. This feature will enable secure, time-limited access to specific files or buckets for uploading and downloading.

# 4. Basic API endpoints

When connecting to Amazon S3, use the following endpoint provided by your account manager: **https://api.mantis -intelligence.com/client -name/batchfiles** . This endpoint will direct your operations to the S3 buckets allocated for your usage.

# 5. Data Buckets Configuration

## 5.1. Input Bucket Configuration (as specified by your Account Manager)

Your Account Manager will provide the name of your input bucket. You need to upload JSON files to this bucket, each containing a list of URLs that you want analysed. The structure of these files should be as follows:

```
□[
  {  "url":     "<url    0>"   },
  {  "url":     "<url    1>"   },
  {  "url":     "<url    N>"   }
]
□
```

## 5.2. Output Bucket Configuration (as specified by your Account Manager)

Similarly, the name of your output bucket, such as publisher-name-output, will be provided by your Account Manager. This is where JSON files containing the safety ratings of the processed URLs will be stored. Each output file will be named according to the following pattern: **<input file name>:<etag from S3>_<date>.json** . This naming convention helps track when and from which input file each output file was generated.

# 6. Processing and Performance

The Mantis Batch Processor can handle a large number of URLs in one or more files. However, for optimal performance and processing time, we recommend a maximum of 1000 URLs per batch file. The larger the file, the longer the processing time, so keeping the number of URLs within this recommended limit helps ensure efficient and timely analysis.

# 7. Output Details

The output files will contain all processed URLs from the input file unless a processing error occurs (see 8.9 for details on failures) . Each file will be uniquely identified by an ETag and timestamp, ensuring easy tracking and management of data outputs.

# 8. Article Brand Safety Classification

When an article is to be analysed, there are two approaches to integration with Mantis:

1. Mantis synchronous HTTP API endpoint
2. Mantis Asynchronous Batch File upload and retrieval

These are explained in more detail below.

## 8.1. Mantis Synchronous HTTP API

Call the Mantis classifyArticle API endpoint for the customer Mantis account (to be provided), e.g.

https://<publisher-mantis-url>/classifyArticle

**POST** article data to this URL as a JSON object in the request body, including:

- HTML content (just the article body content itself, not the whole page with headers/footers/links and teasers for other articles/commercial and other embedded components) or plaintext of the article content.
- Public Page URL
- CMS id (publisher-specific unique article reference id)
- Author(s)
- Title
- Published/last modified timestamps

Pass publisher authentication credentials (to be provided) using an HTTP Basic authentication header. The article will be processed and Brand Safety ratings will be generated and returned from the Mantis API in the response body as JSON data.

To analyse multiple articles at once, the endpoint will accept either a single JSON object describing one file or an array of objects (one for each file to be processed). If an array of article data objects is passed, the response will also be an array of data objects with ratings for each file. However, this HTTP API operates synchronously and will have to process all articles before a response is sent. Therefore, it is only suitable for very small batches of files. For larger batches of files, the asynchronous batch file processing interface should be used (see below).

## 8.2. Mantis Asynchronous Batch File Processing

The data formats used to process larger batches of articles are very similar to those used with the synchronous HTTP API endpoints – containing arrays of JSON data objects (see 8.5) describing the files to be processed and the ratings for each processed article.

To accommodate larger batches, the article data is uploaded as separate files to the Mantis platform.

The primary integration method for file upload uses Amazon S3. A customer-specific storage bucket will be provisioned, and the files containing the article data should be uploaded to this bucket.

Once the files are uploaded, they will be processed by the Mantis analyser, and the results will be stored as new files in the same storage bucket from which they can be downloaded as required.

## 8.3. Article Brand Safety Retrieval

When an article page is built in the publisher CMS backend, the results can be made available on the page by either:

1. Calling the publisher Mantis API endpoint with a GET request passing one of the following as a query parameter:

- Publisher’s CMS article id (recommended)
- Public page URL

2. Retrieving the locally stored ratings data, downloaded from the batch or HTTP API endpoints

When calling the Mantis API endpoint, Brand Safety ratings will be retrieved from the Mantis data store and returned in the response body as JSON data. If the requested article has not yet been processed by Mantis, the JSON response will indicate that the article ratings are not yet available.

For a backend integration, we suggest adding a static script tag to the page that includes the ratings retrieved from the Mantis API as a global object on the page:

```
<script>
  window.mantis    = {ratings    object   from   Mantis};
</script>
```

Publishers can customise this to their own requirements and directly include the ratings data in any code delivered from their CMS backend.

If defined as in the above example, the ratings data from the global mantis object can also be read in any frontend script integrations (e.g., in Ad tags, where we recommend passing a list of ratings values as a mantis key/value pair in page level targeting).

## 8.4. Request and Response JSON Data Formats

### HTTP POST Requests

classifyArticle POST requests should send article data in the request body as a single JSON object, or optionally an array of objects as in the following example:

```
[{
    "html": "<title><h1>FA reveal bans for Port Vale's Mitch Clark and Tom Pope</h1></title><h2> <p>Mitch Clark and Tom Pope were sent off in the closing stages of Port Vale's 3-2 win at Forest Green</p> </h2><p>Port Vale duo Mitch Clark and Tom Pope face three and one game bans respectively.</p> <p>The bans have been confirmed on the FA’s website this morning after both were sent off in last night’s 3-2 win at Forest Green Rovers.</p> ... <p><a data-content-type=\"section-topic\" data-link-tracking=\"InArticle|Link\" href=\"https://www.stokesentinel.co.uk/all-about/port-vale-fc\">For all your latest Port Vale news and analysis, click here</a></p>",
    "cmsID": "stokesentinel-3838610",
    "url": "www.stokesentinel.co.uk/sport/football/port-vale-pope-clark-bans-3838610",
    "author": "Michael Baggaley",
    "published": "2020-02-01T11:27:15.000Z",
    "lastModified": "2020-02-01T11:27:15.000Z",
    "title": "FA reveal bans for Port Vale's Mitch Clark and Tom Pope"
}, {
    "cmsID": "mirror-21482732",
    "url": "www.mirror.co.uk/news/uk-news/inside-lonely-coronavirus-quarantine-masked-21482732",
    "author": "Matthew Dresch",
    "published": "2020-02-12T14:43:27.000Z",
    "title": "Inside lonely coronavirus quarantine where masked Brits have meals left at the door"
}]
□
```

## 8.5. Request Object Details

| Field Name | Required? | Description |
|---|---|---|
| html | One of html or text fields is required. If it is not possible to pass these, then URL is mandatory. | HTML markup of article content (do not include other markup from published pages such as header, footer, links, or teasers for other articles). |

| | | |
|---|---|---|
| text | One of html or text fields is required. If not possible to pass these, then the URL is mandatory. | Plain text version of article content. |
| url | At least one of url or cmsID is required. | URL that will reference the article when it is published. |
| cmsID | At least one of url or cmsID is required. | Publisher defined a unique id for the article (represented as a string value). |
| title | Not mandatory, but useful for reporting and analytics. | The article title. This should also be included in the html or text data to ensure it is processed by the Mantis analyser. |
| author | Not mandatory, but useful for reporting and analytics. | The article author's name. For multiple authors, format as a comma-separated list of names in a single string. |
| published | Not mandatory, but needed for Mantis to understand article versioning. | Timestamp for when the article was published. |
| lastModified | Not mandatory, but needed for Mantis to understand article versioning. | Timestamp for when article content was last changed. |

## 8.6. Batch File Upload Formats

The data format for files uploaded to the batch processor is the same as the request object data format in HTTP endpoint detailed above, but can support much larger numbers of articles included for analysis in a single file.

For customisation, to support other file upload formats please contact us to discuss the requirements.

## 8.7. HTTP GET Requests

classifyArticle GET requests should pass the article URL or CMS id as query parameters in the request, e.g.

```
□https://<publisher - mantis - url>/classifyArticle?c msID=stokesentinel - 3838610
```

□Or

```
□https://<publisher - mantis -
url>/classifyArticle?url=www.stokesentinel.co.uk/sport/football/port - vale - pope -
clark - bans - 3838610
```

□

## 8.8. HTTP GET Requests

Both POST and GET requests will return results in the response body as in the following examples:

**Success:**

```
□{
    "input": {
        "html": "...",
        "cmsID": "mirror - 21482732",
        "url": "www.mirror.co.uk/news/uk - news/inside - lonely - coronavirus - quarantine -
masked- 21482732",
        "author":   "Matthew  Dresch",
        "published":   "2020 - 02- 12T14:43:27.000Z",
        "title":   "Inside  lonely  coronavirus  quarantine  where  masked  Brits  have  meals
left  at  the  door"
    },
    "ratings":   [{
        "customer":   "Default",
        "rating":   "RED",
        "ruleSetVersion":   5
    },  {
        "customer":   "Custom  Ruleset  1",
        "rating":   "AMBER",
        "ruleSetVersion":   5
    },  {
        "customer":   "Custom  Ruleset  1",
        "rating":   "GREEN",
        "ruleSetVersion":   5
    }]
}
```

□

## 8.9. Response Object Details

| Field Name | Description |
|---|---|
| input | Reflects the data that was sent in the original Mantis API request to analyse the article and contains all the article metadata such as: |
| | - cmsID |
| | - URL |
| | - Author |

| | |
|---|---|
| | - Title |
| | - Published timestamp |
| | - Last Modified timestamp |
| | This is in the same format as the POST endpoint request object. |
| ratings | This is an array of ratings using different Mantis rulesets - which can be customised for specific publications, advertising partners, etc. |
| ratings[n].customer | The name of the customer/ruleset this rating is for. |
| ratings[n].rating | The Brand safety rating for the article for this ruleset - values will be “RED”, “GREEN”, “AMBER”. See ratings description below for more details. |
| ratings[n].ruleSetVersion | The version number of the Mantis ruleset that was applied to the article to determine the Brand Safety rating. |
| | Other fields may be present with a more detailed breakdown of the rules that resulted in the brand safety rating – if this has been enabled in the Mantis API platform. |
| | If multiple articles are submitted for processing in a single request, then the response will be an array of objects. |

### Failure:

The response will include an ‘error’ field in the JSON data, with a description of the error that occurred.

```
{
    "error":    "Article    id   not   found"
}
```

□

## 8.10. Ratings Description

The Watson ML system and Mantis Brand Safety Engine work together to determine the Brand safety rating for an article, using a set of customisable Brand Safety rulesets.

When an article is analysed, a variety of semantic data is extracted from the content and is correlated against thresholds defined in the rulesets.

If any individual element is found to exceed the Brand Safety threshold in the ruleset, this will automatically trigger a “RED” rating (Content is NOT brand safe).

If none of the extracted data matches in the ruleset, then this automatically triggers a “GREEN” rating (Content IS brand safe).

Finally, if any of the extracted data is matched in the ruleset, but at less than the configured Brand Safety threshold, the article is given an “AMBER”rating (potentially brand unsafe). How this would be used depends on publisher and advertiser preferences. Ideally, the ruleset would be tuned for a particular purpose, and use of the Amber rating can be used to balance between very conservative brand safety settings and availability of inventory for campaigns (while still ensuring that all unsafe content can be clearly identified).

## 9. Contact Information

### General Inquiries

For general questions or if you are unsure who to contact, please reach out to our support team, and we will direct your inquiry to the appropriate department.

Email: hello@mantissolutions.com

### Client Services

For client support, account management, and service-related inquiries:

Email: clientservices@mantissolutions.com

### Mantis Partnerships

For strategic partnerships and high-level management inquiries:

Ben Pheloung - Head of Mantis

Email: ben.pheloung@mantissolutions.com

### Product & Technical Implementation

For inquiries related to product development, roadmap planning, or technical integration:

Callum McAdam – Senior Technical Product Manager

Email: callum.mcadam@mantissolutions.com

Gloria Bricalli – Product Manager

Email: gloria.bricalli@mantissolutions.com

We value your feedback and are dedicated to ensuring your experience with Mantis meets your expectations. Please do not hesitate to reach out with any concerns or feedback you may have.