**Manifest File**

Each archive of information—which may use 7z, zip, or tar+gzip—will have a manifest file in the archive with the filename manifest.json.

Format:

1. JSON – In stdlib, less reliant on and sensitive to whitespace, more commonly used.

Key Tags:

1. “version” this will start at and default to 1.0. Having this in the file will allow seamless future upgrades.
2. “primary” this will describe the primary file using ONE of the following formats:
   1. “filename” will give the name of the file if it is located in the folder with the manifest.
   2. “hash” will give a cryptographic hash in the form of a CID if it is hosted elsewhere.
3. “additional” will provide a list of additional files, referenced by name in the primary document, each described with the following format:
   1. “name” will provide a unique name for the supporting file.
   2. “type” will provide a format descriptor (initially including UTF8, UTF16, CSV, HTML, GIF, JPEG, MP3, OGG, AV1, and WEBM to support text, formatted text, images, audio, and video).
   3. “filename” will give the name of the file if it is located in the folder, OR
   4. “hash” will give a cryptographic hash in the form of a CID if it is hosted elsewhere.
4. “support” will provide a list of hashes of archives that include files referenced but not included in the “additional” list. Each element will have the following form:
   1. “hash” will give a cryptographic hash in the form of a CID for the supporting archive.
5. “juryParameters” will provide details to the AI node describing how the AI jury should be composed and its decision making process.
   1. NUMBER\_OF\_OUTCOMES: This is the number of possible outcomes that the AI will choose between. For a true/false query, the number of outcomes will be 2.
   2. AI\_NODES: A vector containing {AI\_MODEL, AI\_PROVIDER, NO\_COUNTS, WEIGHT} describing the AI models that will deliberate on the query.
   3. ITERATIONS: The number of iterations to run through the AI jury.

The “primary” file will have a specific format that will be used.

QUERY: This keyword will be used to specify the query for the AI node. If the query directly references supplemental data from the “additional” file, the reference to the name will be prepended with REF: so it’s clear that the file can be obtained by parsing the “additional” section of the manifest.

**Example Manifest Files with Associated Archive Files**

**Example 1: True/False Image Verification**

**Archive Contents:**

• manifest.json

• primary.txt

• IMG\_4872.JPG

• IMG\_4873.JPG

**manifest.json:**

{

"version": "1.0",

"primary": {

"filename": "primary.txt"

},

"additional": [

{

"name": "imageOfWoman1",

"type": "JPEG",

"filename": "IMG\_4872.JPG"

},

{

"name": "imageOfWoman2",

"type": "JPEG",

"filename": "IMG\_4873.JPG"

}

],

"juryParameters": {

"NUMBER\_OF\_OUTCOMES": 2,

"AI\_NODES": [

{

"AI\_MODEL": "GPT-4",

"AI\_PROVIDER": "OpenAI",

"NO\_COUNTS": 3,

"WEIGHT": 1.0

}

],

"ITERATIONS": 1

}

}

**primary.txt:**

QUERY: The woman's dress in these images is red. True or False?

REF:imageOfWoman1

REF:imageOfWoman2

**Example 2: Data Analysis with External Support**

**Archive Contents:**

• manifest.json

• primary.txt

• data.csv

**manifest.json:**

{

"version": "1.0",

"primary": {

"filename": "primary.txt"

},

"additional": [

{

"name": "dataset",

"type": "CSV",

"filename": "data.csv"

}

],

"support": [

{

"hash": "bafybeid7yg3zb76beig63l3x7lxn6kyxyf4gwczp6xkjnju6spj3k2ry6q"

}

],

"juryParameters": {

"NUMBER\_OF\_OUTCOMES": 4,

"AI\_NODES": [

{

"AI\_MODEL": "GPT-4",

"AI\_PROVIDER": "OpenAI",

"NO\_COUNTS": 2,

"WEIGHT": 0.7

},

{

"AI\_MODEL": "BERT",

"AI\_PROVIDER": "Google",

"NO\_COUNTS": 2,

"WEIGHT": 0.3

}

],

"ITERATIONS": 3

}

}

**primary.txt:**

QUERY: Based on the data provided, which factor most significantly impacts sales?

A) Price

B) Marketing Spend

C) Seasonality

D) Product Quality

REF:dataset

**Example 3: Audio File Analysis with External Primary File**

**Archive Contents:**

• manifest.json

• transcript.txt

**manifest.json:**

{

"version": "1.0",

"primary": {

"hash": "bafybeihdwdcefgh4dqkjv67uzcmw7ojee6xedzdetojuzjevtenxquvyku"

},

"additional": [

{

"name": "transcript",

"type": "UTF8",

"filename": "transcript.txt"

}

],

"juryParameters": {

"NUMBER\_OF\_OUTCOMES": 3,

"AI\_NODES": [

{

"AI\_MODEL": "Whisper",

"AI\_PROVIDER": "OpenAI",

"NO\_COUNTS": 5,

"WEIGHT": 0.8

},

{

"AI\_MODEL": "DeepSpeech",

"AI\_PROVIDER": "Mozilla",

"NO\_COUNTS": 2,

"WEIGHT": 0.2

}

],

"ITERATIONS": 2

}

}

**transcript.txt:**

REF:audioFile

QUERY: Transcribe the provided audio file and determine the speaker's sentiment (Positive, Neutral, Negative).

**Notes:**

• The primary audio file is not included in the archive but is referenced via its hash.

• The transcript.txt serves as an additional file containing the query and reference to the audio file.

**Explanation of Fields:**

• **version**: Indicates the manifest version for compatibility.

• **primary**: Contains information about the primary file, either via filename if included or hash if external.

• **additional**: Lists supplementary files required for the query, each with a unique name, type, and filename or hash.

• **support**: References external archives containing additional files not included in the additional list.

• **juryParameters**: Defines how the AI jury is composed and operates, including the number of outcomes, AI models used, counts, weights, and iterations.

**Conclusion**

The manifest format is comprehensive and effectively supports the evaluator function’s requirements. By ensuring consistent key naming and clarifying optional versus required fields, the format can be further optimized for ease of use and integration.