

Azure AI Services Integration within TTA Questions

000045581 | General Info | Last updated 2026-09-11 | TotalAgility

ISSUE

Customer has forwarded several questions regarding Azure AI services integration.

SOLUTION

For OCR, can you confirm what the Base64 payload represents (e.g. complete document content only, or does it include metadata, document identifiers, user information, or other attributes)?

>> This is more towards how Microsoft Azure works and not so much TotalAgility. Microsoft's Azure AI Document Intelligence REST specification defines base64Source as the "Base64 encoding of the document to analyze." The documented request body contains either base64Source or urlSource; it does not define document identifiers, workflow IDs, usernames, or similar business metadata as properties of that Base64 value.

[Document Models - Analyze Document - REST API \(Azure Azure AI Services\) | Microsoft Learn](#)

I have run a test with Fiddler with a sample tiff from the Capture Started Pack sample images (Order 4).

The base64Source in the JSON request decoded to a single-page PNG image with dimensions of 1696 × 2200 pixels at approximately 200 DPI. The supplied source was a single-page TIFF with the same dimensions and resolution. A pixel-level comparison confirmed that the decoded PNG contains the complete visual content of the TIFF page, with no cropping, resizing, rotation, or other pixel changes.

The original TIFF file itself is not transmitted in Base64Source. Instead, its page is converted to PNG and the resulting PNG binary is Base64-encoded.

The decoded PNG contains only its image header, resolution information, compressed pixel data, and end marker. It contains no text or EXIF metadata sections, and we found no TotalAgility document identifiers, job identifiers, workflow information, usernames, file names, or other customer attributes embedded in the image payload. Thus confirming the above statement linked to the Microsoft Azure article.

AEL, can you confirm the exact content/structure of the JSON payload sent to Azure (e.g. text only, prompts, metadata, identifiers)?

>> JSON request structure

```
{
  "MessagePrompts": [
    {
      "Role": "user",
```

```
"ContentCollection": [  
  {  
    "MessageType": "text",  
    "MessageText": "<combined extraction prompt>"  
  }  
]  
},  
"Tools": null,  
"Data": "<opaque hexadecimal value>"  
}
```

The text prompt contains (current scenario):

1.Static AEL extraction instructions

- How to interpret OCR text.
- How text segments and their sequence should be interpreted.
- How checked and unchecked options are represented as [TRUE] and [FALSE].
- How extracted fields and supporting segment IDs must be returned.

2.A worked extraction example

- Example document text.
- Example field schema.
- Example output JSON.

3.The actual OCR-derived document text

- All 49 indexed text segments from the tested document.
- Nested arrays representing related groups of text.
- Names, addresses, products, prices, totals, payment selections, signature text, and dates appearing in the document.

4.The field schema

- FieldIndex
- FieldName
- FieldDefinition

5.Output instructions

- The model is instructed to return only the required JSON result.
- It must include the extracted value and the OCR text-segment index supporting it.

Can you provide more details about the content of the returned Azure responses (e.g. what information is contained in AnalyzeResults and Prompt Results)?

>> The completed AnalyzeResults response contains the recognized document information, including:

- The complete OCR-recognized text.
- Page number, page dimensions, orientation and measurement unit.
- Recognized words and lines.
- Paragraph groupings.
- The location of text on the page, represented by polygon coordinates.
- Recognition-confidence values for individual words.
- Text offsets and lengths linking each word, line and paragraph to the combined OCR text.
- Handwriting classifications and associated confidence values where handwriting was detected.
- The Azure API version, recognition model and processing timestamps.

The response does not contain the original TIFF or PNG image. However, because it includes the complete recognized text, it can contain business and personal information appearing in the document, such as names, addresses, monetary values, dates and other recognized content.

The complete Prompt Result also contains service-level information, including:

- The model used: gpt-4.1-mini-2025-04-14 (in my test).
- A service-generated chat-completion identifier.
- The completion timestamp.
- Input, cached-input and output token counts.
- The completion finish reason.
- Prompt and response content-filter results.
- Service-tier and system-fingerprint values.
- Latency and processing measurements.
- Error, tool-call and processing-status fields.

The Prompt Result does not return the original document image and does not echo the complete OCR text or full prompt. Those are present in the AEL request, while the response contains the requested extracted values, their supporting OCR segment identifiers and model-processing metadata.

When stating that the interaction is only visible through Fiddler, does this mean that TTA provides no native logging, monitoring, audit trail, or UI visibility for these Azure calls?

As confirmed by TS expertise in the previous case (28164019), unless you are tracing with Fiddler, you cannot see the interaction.

Are Azure OCR/AEL interactions logged anywhere (e.g. TTA logs, Azure logs, Application Insights, or any other logging mechanism)?:

As previously mentioned by TS expertise:

Fiddler has been the only way that I have found to trace OCR/AEL calls.

If there is an issue that needs to be debugged, please advise what it is so we can test locally.

If we have some context to the reason behind the queries, we would be in a better position to answer.

Additionally, if a OCR call fails with hard error, a fallback OCR engine can be set on project level. If a AEL call fails with hard error, an error is logged against the job. To ensure quality of the results provided by the AEL (and other locators), we recommend using validation/ validators, e.g. checking if a field is blank.

Furthermore, kindly note the following entry that aims to improve AI services within TotalAgility:

Bug 2216355:[FoX] Implement a retry mechanism in Transformation Server when communicating with Gen AI services

Currently, the Transformation Server performs 3 retries at job level in the instance of an error.

However, there also needs to be a retry implemented at component level when communicating with Gen AI services via the AEL, Azure OCR etc.

This would ensure the Transformation Server is consistent with how the TTA App and TTA Core Worker components perform retries for Gen AI services.

Release: TA 2026.3

When stating that no Azure-related identifiers are available, does this mean that request IDs, operation IDs, or correlation IDs are not generated, not stored, or simply not exposed to the customer?:

As previously noted in case 28164019, kindly note the following:

“Not visible” should be interpreted as not exposed through the customer-accessible TTA request, response, or client-side trace, rather than as confirmation that correlation identifiers are not generated or stored anywhere.

Our Fiddler captures show that Azure-related identifiers are generated and returned in some responses:

The OCR response contains an Azure Document Intelligence resultId, including an analyzeResults operation GUID.

The AEL response contains an Azure OpenAI-style chat-completion identifier, such as a chatcmpl-... ID.

However, we did not identify a TTA document-processing request ID, transaction ID, or correlation ID that is exposed alongside those Azure identifiers and could be used by the customer to map:

TTA request ↔ Tungsten Cloud request ↔ Azure AI operation

Fiddler only captures the communication between the TTA environment and the Tungsten Cloud endpoint. It cannot capture the subsequent server-side communication between Tungsten Cloud and Azure. Therefore, the trace cannot show whether Tungsten Cloud creates or stores an internal mapping between its own transaction and the Azure request or operation ID.

In conclusion, TTA uses Azure AI through Tungsten Cloud, and Azure generates identifiers for the downstream operations. However, there is currently no exposed customer-facing reference that allows the customer to correlate a specific TTA transaction bidirectionally with the corresponding Azure AI request.

Additional questions:

1/ Which requests were submitted to Azure for Document X?

They use OCR and AutoExtract Locator, which calls their own BYOL.

TotalAgility will make requests under the hood to the AI Provider, for OCR to be performed.

2/ What results were returned by Azure for each document?

OCR will be returned

3/ Which logs exist for these transactions?

We have already provided full details for Audit Transactions which are logged in TotalAgility.

4/ Which references, identifiers, or Correlation IDs are available?

Within the TotalAgility product, these items are not stored.

5/ Can a specific transaction be traced or reconstructed at a later point in time?

no, this is not something which is available OOTB.

6/ What audit capabilities are available for this purpose?

The following Audit_Types have been added to log Audit Data for AI functions within the product:

Each of these will have detail in the Audit. For example I called a Generative AI activity:

Testing this at run time we can see the following is written to the Audit Log:

This is the AUDIT_ENTRY_DESCRIPTION, where we can see details such as the Provider and the Token cost:

"Execute AI prompt - The Generative AI activity has been executed. Provider: Tungsten. LLM tokens used: 75"

This example I have given is just one of the additional audit types which I have mentioned above.

7/ Are Azure-related service interactions logged by TTA?

Other than what we store in the Audit log (full details previously provided), nothing is logged.

8/ What specific information is stored in this process?

Please see my previous comment regarding the Audit Log.

9/ Where is this information stored?

The main TotalAgility Database, or the Audit DB if using a split Database.

10/ Which user groups and/or roles have access to it?

These can be viewed directly in the database (via SSMS) or via the Workspace in the OOTB ViewAuditLogEntries.form.

11/ How can a specific processing transaction be traced or reconstructed retrospectively?

This cannot be performed manually

12/ Are Azure-related identifiers stored internally?

These items are not logged or stored in TotalAgility.

<https://aio-eus-uat-cae-aif-app14-local.redglacier-35d7ee4f.eastus.azurecontainerapps.io/article/45581>