Artificial Genius *Orthix:*

Non-Hallucinatory Agentic Unstructured Data Workflow Platform

Customer Manual

Version AWS.1.2 (June 23, 2026)

How to Unlock Commercial Returns from Unstructured Data

Enterprises have a lot of ***valuable unstructured data*** which, if ***the accurate and relevant information hidden in it*** could be delivered to the ***right person at the right time*** – would create significant opportunities for sales, revenue, and other ***measurable commercial returns***.

Making It Possible: A New Generation of Language Models

Achieving the until now elusive combination of reliability, accuracy, and relevance required for enterprise use of language models necessitates a generational shift in how language models are created and applied to obtain the maximum commercial value from their capabilities. A perspective on the history of language models explains the need for a new paradigm:

First Generation (1950s): Researchers used symbolic logic to build deterministic, rule-based models. While safe, these models lacked fluency and could not scale.

Second Generation (1980s–present): The shift to probabilistic models (culminating in the Transformer architecture) unlocked incredible fluency. However, because these models predict the next token based on probability, they suffer from unbounded failure modes (hallucinations) that are difficult to engineer away.

Third Generation (Artificial Genius approach): Rather than a new generation that replaces the old, we're moving from the rigidity of symbolic logic and the unpredictability of probabilistic models toward a hybrid architecture. This approach uses the generative power of Second-Generation base models to understand context but applies a deterministic layer to verify and produce output. It's the convergence of fluency and factuality.

It's mathematically difficult to prevent standard generative models from hallucinating because the extrapolative, generative process itself causes errors. Artificial Genius addresses this by using the model strictly ***non-generatively***. In this paradigm, the vast probability information learned by the model is used only interpolatively on the input. This allows the model to comprehend the innumerable ways a piece of information or a question can be expressed

without relying on probability to generate the answer. To create this third-generation capability, Artificial Genius post-trains existing Second-Generation base models (Amazon Nova).

This patented method ***intelligently removes the output probabilities from the model***. While standard solutions attempt to ensure determinism by lowering the temperature to zero (which often fails to address the core hallucination issue), Artificial Genius post-trains the model to tilt log-probabilities of next-token predictions toward absolute ones or zeros. This fine-tuning forces the model to follow a single system instruction: don't make up answers that don't exist.

This creates a mathematical loophole where the model retains its genius-level understanding of data but operates with the safety profile required for finance and healthcare.

Orthix Non-Hallucinatory Foundation Models

Artificial Genius' third-generation, non-hallucinatory foundation models are built on the base of the leading Amazon Nova family of second-generation models, currently, ***Amazon Nova Lite 1.0.*** The terms of your license to the Orthix models incorporate the terms of the Amazon Nova license.

Artificial Genius has gone beyond solving the original hallucination problem, to understand the broader contextual needs for reliability in the enterprise, and is therefore currently offering two language models on the platform:

Orthix-30303: this is the original ***pure non-hallucinatory*** language model with 0.03% hallucination rate. It has been trained to distinguish strictly answerable from strictly non-answerable non-generative questions. In practice, when given questions with some degree of ambiguity, will mostly refuse to answer them. It is application-dependent whether this is too strict a notion of non-hallucination.

Orthix-33333-10-Pct: this is a new language model that has been trained to distinguish non-generative questions that are either answerable or non-answerable, up to a ***controlled amount of ambiguity*** that would be contextually resolvable by a human. This model has a hallucination rate of 0.6%, not counting as hallucinations answers that would be contextually correct based on human judgment despite the controlled amount of logical ambiguity.

When purchasing the ***Artificial Genius Orthix*** on AWS Marketplace, Artificial Genius will also enable you to obtain access to these non-hallucinatory foundation models via ***Bedrock Custom Model Import***. Please contact your AWS representative to enable this import. Once imported, you will need to note the resulting ***model deployment ARN(s)*** as input to the workflow platform.

What is a Non-Generative Question?

It is important to understand that these Third-Generation language models are ***not chatbots***. To work around the impossibility of removing hallucinations, these models are designed to reliably answer ***non-generative questions***. A non-generative question is one for which the model ***wouldn't need to use any probabilities***, in principle, to generate the sequence of tokens for the answer, but rather could extract or verify information based solely on the input context.

While short answers (such as dates or names) are obviously non-generative, it's also possible to output long sequences deterministically as well. For example, asking for a direct quote from a document to justify a previous answer is a non-generative task.

We'll show some examples of both answerable and non-answerable, non-generative questions, regarding the following input context:

> **Context**: "Financial performance remained strong through the third quarter. Our revenue grew by 15% year-over-year, driven by robust sales in the enterprise segment."

Answerable, non-generative, short answer:

> **Question**: "What was the annual revenue growth?"
>
> **Answer**: "15%"

Answerable, non-generative, long-answer, follow-up question, based on the answer to the previous one:

> **Question**: "Provide a quote from the document showing that the annual revenue growth was 15%."
>
> **Answer**: '"Our revenue grew by 15% year-over-year, driven by robust sales in the enterprise segment."'

Unanswerable, short-answer question:

> **Question**: "What was the CEO's bonus this year?"
>
> **Answer**: "Unknown"

Orthix Enterprise Workflow Platform

Enterprise workflows ***process unstructured data*** from selected input databases or feeds, to produce and ***disseminate unstructured data*** as entries or reports in output databases or feeds.

On the ***Orthix*** platform, Artificial Genius' third-generation language non-hallucinatory foundation models can be used to make these workflows safe, reliable, and domain-relevant. The method is to decompose the information processing step into a set of hallucination-free,

non-generative extractions from the input data, followed by ***reassembly*** of these extracted items into the final report. When the number of input sources is large and stored in a document database, the extracted items are precomputed and stored in an ordinary relational database.

Orthix enables each domain-specialized department to create their own workflows on demand, using a special semi-structured prompt.

Product Requirements Document (PRD) is an industry standard methodology for specifying a desired software artifact. In Artificial Genius' platform, the agent accepts the PRD in the form of a ***semi-structured prompt*** in the industry-standard ***Markdown*** format to specify the workflow. This PRD includes the specification of:

- Non-hallucinatory language model;
- Input source of unstructured data;
- Output sinks for the resulting structured or unstructured data;
- Items of information to be extracted non-generatively from the inputs;
- A template for the reassembly of the extracted information into a report.

Each domain-specialized department can be supplied with suitable PRD templates by the AI leadership that enable them to use the ***Orthix*** platform in a self-service manner to create custom, domain-specific workflows on demand.

In more detail, the PRD is a document in Markdown format. This PRD may include Markdown link definitions, some of which are reserved by the PRD for the workflow platform. The ***format of labels*** for such reserved link definitions takes the form ***Dash-Separated-Capitalized-Words-or-Numbers***, including:

- A reserved set of labels (**"Name"**, **"Agent"**, **"Input"**, **"Output"**) that are interpreted by the workflow platform.
- Labels **"-…-"**, starting and ending with a dash, that are interpreted by the Agent client (minus the enclosing dashes).
- Labels **"-…"**, starting with a dash, that are interpreted by the Input server (minus the starting dash).
- Labels **"…-"**, ending with a dash, that are interpreted by the Output server (minus the ending dash).
- Labels starting with the letter **"Q"**, followed by a series of ***one or more dash-separated numbers***, the final number or which may optionally be replaced by a ***wildcard "?"***, which are interpreted as non-generative questions to ask the agent about the input to produce the output, in a hierarchically numbered order.

More specifically, the following reserved link definitions are used within the PRD to configure the workflow, and the defined values of these reserved links can be subsequently incorporated into other link definitions, or into the workflow output, via reserved link references to the corresponding label:

[Name]: specifies the name of the workflow.

[Agent]: specifies the non-hallucinatory language model, as an ARN.

[-Field-]: specifies parameters for the Agent, surrounded by dashes. Currently available Agent field labels include ***-Region-***, ***-Tries-***, ***-Timeout-***, and ***-Max-Tokens-***.

[Input]: specifies the data input server according to ***Apache OpenDAL*** standards. Currently whitelisted input servers are ***S3*** (Amazon Simple Storage Service), ***HTTP*** (Web server), and ***FS*** (local filesystem, internal to the platform container, for test purposes). The workflow will ***process all available documents*** on the specified input server; however, the scope may be narrowed by specifying a ***Root-*** on that server as described below.

[Field-]: specifies any parameters necessary to set up the connection with the input server, according to Apache OpenDAL standards:

- ***S3***: the ***Region-*** and ***Bucket-*** names must be specified. For S3 buckets not owned by the workflow deployment, temporary access to the filesystem for ***Timeout-*** seconds (by default 3600) can be obtained by specifying an IAM ***Role-*** (as an ARN) that owns the S3 bucket, and may be assumed by the workflow deployment. A ***Root-*** within the filesystem on the bucket may also be specified to narrow the scope.
- ***HTTP***: the ***Endpoint-*** base URL of the website must be specified. Because HTTP servers do not offer a way to automatically list all the files on the server, in this case, a ***List-*** of space-separated files to be processed must also be specified. A ***Root-*** within the filesystem on that website may also be specified to narrow the scope.
- ***FS***: a ***Root-*** within the local filesystem in the container may be specified to narrow the scope. For security, this ***Root-*** is restricted to be relative to the part of the local filesystem accessible to the ***Agent***, not to the entire local filesystem in the container.
- **[Path-]:** for all input server types, an implicit **[Path-]** field will be supplied to the PRD to designate the specific file that is being processed, relative to the server's **[Root-].**

[Div-]: if provided, specifies the method to split up the documents retrieved from the input server into self-contained parts to which the workflow will be applied. Currently supported splitting methods are ***h1***, ***h2***, …, etc., for splitting up documents at header levels 1, 2, …, etc., for a variety of document formats; and ***mailbox,*** for documents containing a sequence of email messages.

When documents are split, an implicit **[Part-]** field will be supplied to the PRD to designate the specific title of the part of the document that is being processed. In the case of mailbox splitting, an implicit **[From-]** field will also be supplied.

[Output]: specifies the data output server according to Apache OpenDAL standards. Currently, the same server options are whitelisted as for the input servers.

[-Field]: specifies any parameters necessary to set up the connection with the output server, according to Apache OpenDAL standards. These parameters are the same as noted above for the input servers, except that the qualifying dash must be placed at the beginning instead of end of the field name.

[Q1]:, [Q2]:, [Q2-1]:, … specify the ***non-generative questions*** that the agent will answer about each input document (or specified part of a document) to produce the output. These questions are hierarchically numbered according to the dash-separated numerical parts of their labels and will be asked in the corresponding order.

The link definition for a question may consist of general Markdown, including link references, and reserved link references will be entirely substituted with their values prior to rendering the final Markdown that will be passed to the Agent to obtain the answer.

Chained questions are therefore supported: each question may contain reserved link references to prior questions in the hierarchical ordering, which will be substituted by their answers, before the question containing the link is asked. If any such linked question is unanswerable, the question linking to it will also be deemed unanswerable. For example,

```
[Q1]: What is the name of the company this Document is about?
[Q2]: What is the annual revenue of [Q1]?
```

Moreover, unbounded chains of questions may be created using link definitions that include wildcards in the label. For example,

```
[Q3-1]: What is the name of a customer of the company?
[Q3-?]: Other than any of [Q3-?], what is the name of a
DIFFERENT customer of the company?
```

enables an arbitrarily long list of customer names to be extract from the Document, producing corresponding link labels [Q2-1], [Q3-2], … Because questions are always asked in hierarchical order, when each wildcard question is being asked, only the prior matches to the wildcard exist to be substituted into the current wildcard question.

[Inf]: specifies the maximum number of wildcard iterations when generating questions. The default value is 12.

Markdown text that will form the ***output document*** corresponding to each input document.

This Markdown may contain regular link references, like `[link text][Q1]`, or shortcut link references like `[Q1]`, to the above reserved link definitions. Reserved link references,

unlike ordinary, “lazily” followed, Markdown link references, will be “eagerly” substituted by their link texts – which are by default their link values – prior to rendering the final Markdown output. The value of any reserved link reference to a question is the answer to that question, if answerable – not the question itself.

For a reserved link reference that includes explicit link text instead of the default, the link text may as usual consist of general Markdown (including further link references), within which the link value may be substituted with the special link reference **“[_]”**, after which any further link text is ignored (except for wildcards, see below). As noted, the default link text of a reserved link reference is its link value (not its link label). Thus, the two reserved link references below are equivalent, and will be substituted with the link value:

```
[label]
[[_]][label]
```

If the value of a reserved link is undefined (for example an unanswerable question), then the entire link text will be omitted in the substitution, whether or not it is specified to include the (undefined) link value. Thus, if question [Q1] is either answerable with answer “A1”, or unanswerable, then the reserved link:

```
Checking: [the answer is [_]][Q1]
```

will be substituted with the first or second text below, respectively:

```
Checking: the answer is A1
Checking:
```

If a reserved link reference label includes a wildcard, then the values of all the reserved links whose labels match that wildcard, and have defined values, will be substituted, in order, into copies of the link text, except that the part of the link text after the link value substitution (**“[_]”**) will be omitted for the final substituted link. Thus, if there are questions [Q3-1] and [Q3-2], with answers “A3_1” and “A3_2”, then the two reserved links below are equivalent:

```
[Q3-?]
[[_], ][Q3-?]
```

and either will be substituted with:

```
A3_1, A3_2
```

The specific value of the wildcard itself may be substituted into the link text with the special link reference **“[?]”.** Thus, the reserved link:

```
[[?]. [_]; ][Q3-?].
```

will be substituted with the Markdown numbered list:

```
1. A3_1; 2. A3_2.
```

Example Product Requirements Document (PRD)

We'll now run through an example PRD that uses the open-source "***enronsent***" unstructured dataset, consisting of emails between Enron officers and employees leading up to the company's collapse. The goal of the workflow is to establish a communication graph between the various personnel.

```
[Name]: Enron Analysis
[Agent]: arn:aws:bedrock:us-east-1:000000000000:custom-model-
import/9if93r4if04i
[-Region-]: us-east-1
[Input]: fs
[Root-]: enronsent
[Div-]: mailbox
[Output]: email
[-To]: audit@sec.gov
[-Subject]: Enron Communication Analysis: [Part-]
[Q1]: To whom did the author of the Document say something?
[Q2]: Provide a quote from the text showing what the author of the
Document said to [Q1].
[Q3]: Who said something to the author of the Document?
[Q4]: Provide a quote from the text showing what [Q3] said to the
author of the Document.

# Communication Analysis

[[From-] said [Q2] to [Q1].][Q2]

[[Q3] said [Q4] to [From-].][Q4]
```

This example PRD template is available for download at: https://agipub.s3.us-east-1.amazonaws.com/artificial-genius-orthix-prd-example.md . Note that this PRD uses selected full link references with link text, in order that the production of any output is conditional on [Q2] or [Q4] being answerable.

Now we'll run this PRD through the Orthix platform, on some actual entries in the database, using the ***Orthix-30303*** third-generation foundation model. This demo includes some entries where the PRD questions are unanswerable, and so would normally risk hallucinations with a second-generation language model.

Document 1

```
I just spoke with Clark Smith, head of Elpaso's merchant arm. I told
him that we had been hearing that El Paso was blaming Enrononline for
problems in Western gas markets. He asked for some more specifics about
who exactly was spreading the rumor (I told him we had heard it from 3-
```

4 sources). He acknowledged that EOL was not the problem; said he couldn't believe that it had been identified as such; and said he would bring it up on his call with his Washington team this afternoon.
I think he will put it to rest (except for whatever damage has already been done). I did promise to get some more specifics on who has told us that El Paso pointed to us. Can anybody give some info on that?

Questions 1

[Q1]: Clark Smith
[Q2]: That El Paso was blaming Enrononline for problems in Western gas markets
[Q3]: Clark Smith
[Q4]: He would bring it up on his call with his Washington team.

Answers 1

Communication Analysis

Steven J Kean said that El Paso was blaming Enrononline for problems in Western gas markets to Clark Smith.

Clark Smith said that he would bring it up on his call with his Washington team to Steven J Kean.

Document 2

Dear Mr. Kean,
Heidi Van Genderen asked me to send you a reminder notice. We will be loading all the power point presentations onto 1 laptop. If you could please send me your presentation by friday morning (5/18), it would be greatly appreciated.
Christine Velez Badar The Centers at University of Colorado-Denver 1445 Market St., Suite 380 Denver, CO 80202 phone: 303.820.5674 fax: 303.820.5656 email: cbadar@carbon.cudenver.edu

Questions 2

[Q1]: Mr. Kean
[Q2]: “We will be loading all the power point presentations onto 1 laptop.”
[Q3]: Heidi Van Genderen
[Q4]: “Heidi Van Genderen asked me to send you a reminder notice.”

Answers 2

Communication Analysis

Christine Velez Badar said that We will be loading all the power point presentations onto 1 laptop to Mr. Kean.

Heidi Van Genderen said that Heidi Van Genderen asked me to send you a reminder notice to Christine Velez Badar.

Document 3

```
Attached is a draft of the letter we'd like to send to our 16,000
residential customers on Friday. Please review and let me know your
comments by 12 noon on Wednesday.
```

Questions 3

```
[Q1]: Unknown
[Q2]: <blank>
[Q3]: Unknown
[Q4]: <blank>
```

Answers 3

```
Communication Analysis

<blank>
```

Example Enterprise Use Cases

Summarization

Portfolio managers need timely access to key information from analyst reports as soon as they are published. The volume and length of analyst reports is such that a portfolio manager does not have time to read all of them carefully.

Instead, the portfolio manager would like concise and timely summaries of the reports, targeted to their investment objectives. Asking a chatbot to “summarize” the reports would result in irrelevant and hallucinated information.

So the solution is to prepare a PRD to prompt the ***Orthix*** platform to create a workflow to deliver relevant summaries of the reports to the portfolio manager in a timely manner. The PRD specifies:

- The data feed for the analyst reports.
- The specific items of information that might or might not be in an analyst report, that are relevant to the portfolio manager’s objectives.
- A template to reassemble the specific items of information into a well formatted summary.

- The delivery channel for the formatted summary.

Search

A representative fields numerous requests for proposal (RFPs) from clients with particular investment objectives, who would like an investment plan, with specific portfolio assets, entry and exit triggers, etc., justified by the client's investment objectives.

In many cases, elements of the client's investment objectives may be similar to previous clients, for whom an investment plan has been prepared. To increase productivity, representatives would like an AI agent to prepare a draft investment plan in response to an RFP, based on the plans prepared for similar clients, that the representative can use as a starting point, and save hours of research time.

The solution is for the representative's department to prompt the ***Orthix*** platform to create a workflow to input new RFPs, and deliver relevant investment plan drafts to representative in a timely manner. This involves two PRDs, one to analyze the investment plans, and the other to analyze the RFPs.

The investment plan PRD specifies:

- The email inbox to which the investment plans sent to clients are CC'd.
- The specific items of information which characterize the elements of the investment plan that are responsive to specific client investment objectives, which are used to label the elements of the plan.
- The relational database in which these labeled items should be stored.

The RFP PRD specifies:

- The email inbox in which the RFPs are received.
- The specific items of information from the RFP that characterize the client's investment objectives.
- The relational database to search for the investment plan elements, labeled by those items characterizing the investment objectives.
- The template to reassemble the labeled investment plan elements into a draft investment plan.
- The email for the appropriate representative to receive the draft investment plan.

Product Configuration and Security

Platform Configuration

The ***Orthix*** platform is delivered as an AWS Marketplace container product, which therefore runs entirely in your customer AWS account, with no dependencies on Artificial Genius. This container meets all of the stringent enterprise security requirements for the AWS Marketplace.

A typical deployment of the container in your customer account would be as a Service in the ***Elastic Container Service (ECS).*** The container has been built cross-platform, and can therefore be run on either ***x86_64*** or ***arm64*** instances on ECS. The ***Orthix*** platform container requires access to the following resources:

Bedrock Runtime (port 8080): this service is used to access the ***Orthix*** non-hallucinatory language models, and to perform actions as the platform container's agent.

S3 (ports 80, 443): this service is used to access unstructured data stores in AWS S3 buckets.

HTTP (ports 80, 443): this service is used to access unstructured data on the private or public Web.

To support this access:

- The above ports should be specified in the `portMappings` of the ECS Task specification used to launch the Service.
- In your ***Identity and Access Management (IAM)*** configuration, the Task/TaskExecution roles specified in the ECS Task specification will correspondingly need permission either to access the above necessary services, or to be trusted to assume the roles of principals who do. The ECS deployment is recommended to use `MANAGED_INSTANCES` orchestration so that the set of roles that may need such IAM trust relationships is fixed.
- To support temporary access keys for these services, which is more secure than using fixed access keys, the IAM roles involved will also need permission to access the ***Security Token Service (STS).***
- Your VPC or other ***network settings*** will also need to be configured to permit connectivity to any of these services that live in different parts of the network, or if selected public Web sites are to be whitelisted.
- For compatibility with the container's ***Bedrock AgentCore*** runtime, the ECS Task specification should specify the container's `healthCheck` as a `/ping` request on port `8080`.

An ECS Task specification template is available at `https://agipub.s3.us-east-1.amazonaws.com/artificial-genius-orthix-ecs-task.json`.

Model Configuration

The ***Orthix*** non-hallucinatory language models are delivered via ***Bedrock Custom Model Import***. This is a secure service that provides inference endpoints within your AWS account, by hosting and running the Orthix models in a secure, AWS-managed escrow account that:

1. Does not allow Artificial Genius any access to the prompts or responses that you send to the models;
2. Does not allow you any access to the proprietary weights of either the Artificial Genius Orthix models, nor the underlying Amazon Nova models.

Once the import is complete, the models will be accessible to you via ***model deployment ARN(s)***. You will need to provide these ARN(s) to the teams using the platform, which will be entered into their PRD prompts in the **[Agent]:** field.

Running a Workflow

To run a workflow, the ***Orthix*** platform uses the standard container endpoint for ***Bedrock AgentCore*** agents, which is the REST endpoint `/invocations` on port `8080`.

The POST input for this endpoint is JSON of the format `{"prompt"="…"}` containing the PRD in Markdown format, and the POST output is a JSON formatted log of successes and failures of workflow items.

The actual data inputs and outputs are handled by servers as specified in the PRD. In particular, the results of running the workflow are written to corresponding files/entries on the `[Output]:` server specified in the PRD.