

VANTIQ



VANTIQ 1.40 Update

VANTIQ Global Field & Partner Enablement

Agenda

- Overview of New Features and Enhancements
- GenAI Builder - Content Ingestion
- GenAI Builder Enhancements
- LLM Playground Enhancements
- Design Model Enhancements - Conversational Generation
- Server Enhancements
- Client Builder Enhancements
- IDE Enhancements



Overview of New Features and Enhancements

Integration Enhancements - GenAI Builder Content Ingestion

Increased developer productivity

- Content Ingestion Overview
 - Before 1.40 we could ingest content into our Semantic Index by uploading documents into semantic index
 - These capabilities were basic and had some limitation
 - In 1.40 the GenAI Builder has been extended to add a collection of Content Ingestion tasks
 - 1.40 also makes use of the Unstructured library to perform advanced content ingestion. UnstructuredURL is running as a new service inside Vantiq Cluster

Usability Enhancements - GenAI enhancements

Increased developer productivity

- Auditing of GenAI Tasks
 - The GenAI Builder now allows you to tag tasks as being audited
 - This allows you to keep an audit trail of GenAI activities
- Badging and Tracing
- New GenAI Builder capabilities
 - Fallback
 - Tool use
- External Semantic Index Support

Public GenAI Procedure

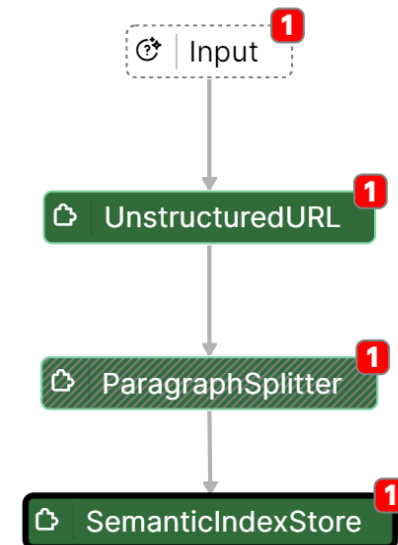
Name: Combined

Return Type: Any | Scalar

Generated Script: [Click to View](#)

Task Tracing: ☒

Task	Audit	Trace
Input	☐	☐
UnstructuredURL	☒	☐
ParagraphSplitter	☒	☐
SemanticIndexStore	☒	☐



Name: ParagraphSplitter

Configuration: [Click to Edit](#)

Description:

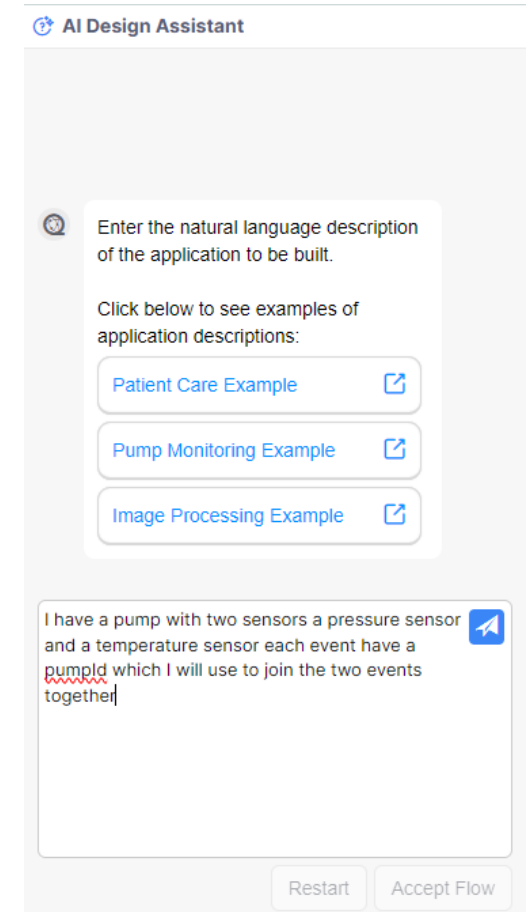
[View Task Output](#)

Δ T	Data
365.4sec	["task":"ParagraphSplitter","output":]

Usability Enhancements - Design Modeler Enhancements

Increased developer productivity

- Change to the AI Design Assistant within the Design Model -AI Design Assistant is now conversational
 - Rather than resubmitting the text, you submit additional follow up message that will change the Flow
 - Slight UI changes to reflect the more conversational nature of the interface





GenAI Builder Content Ingestion

GenAI Builder Content Ingestion Overview

- A number of capabilities have been added to the GenAI Builder and server to support ingestion, transformation, and storing of content into a Semantic Index
 - Document Loaders
 - Two Document Loaders have been added - UnstructuredURL which is integrated with the UnstructuredURL library and the ConversationMemory that converts conversational memory into documents
 - Document Transformers
 - Series of document transformers that split different text-based documents into chunked documents
 - Storage
 - SemanticIndexStore loads a series of documents in a Semantic Index

UnstructuredURL Document Loaders

- UnstructuredURL - make use of the Unstructured library - <https://unstructured.io/> - to convert various text and binary formatted documents into Documents
 - Unstructured provides more advanced document conversation capabilities
 - Does not replace the existing capabilities but should be used when the more advanced capabilities are useful
- The Unstructured Library has been packaged as a new Service within the VantIQ Cluster. The UnstructuredURL document loader integrates with this service.

UnstructuredURL Document Loaders cont.

Parameter	Value Type	Value
input	Array	<code>[{"type":"String","value":"https://test.vantiq.com/ui/docs/NS/mejfm_139..."}]</code> ☐ Use Null
config	Object	<code>{}</code> ☐ Use Null

Array of Url's

Input

UnstructuredURL

Specify initial configuration properties for
io.vantiq.ai.components.loaders.UnstructuredURL Task
'UnstructuredURL'

Reads content from the given url(s) and constructs a list of Document instances.

Optional Parameter	Value
mode (Enumerated)	paged The processing mode used when constructing Document instances. Determines how the content is divided into Documents.
encoding (String)	utf8 The encoding method used to decode the text input.
include_page_breaks (Boolean)	☐ If True, the output will include page breaks if the filetype supports it.
strategy (Enumerated)	fast The strategy to use for partitioning PDF or image content.
languages (Array of String)	<null> The languages present in the document, for use in partitioning and/or OCR. More than one language means that any of them may appear in the document.
skip_infer_table_types (Array of String)	<null> The document types (pdf, png, etc...) for which you want to skip table extraction.
coordinates (Boolean)	☐ If true, return coordinates for each element (when present).
extract_image_block_types (Array of String)	<null> The types of elements (e.g. 'table', 'image') to extract, for use in extracting image blocks as base64 encoded data stored in metadata fields
unique_element_ids (Boolean)	☐ When 'True', assign UUIDs to element IDs, which guarantees their uniqueness (useful when using them as primary keys). Otherwise a SHA-256 of element text is used.

Cancel OK

Return Array of Document's each
represents a single page

```
[{"id": null, "metadata": {"source": {"category_depth": 0, "filename": "VANTIQIntegrationArchitecture.docx", "page_number": 1}, "languages": {"filetype": "application/vnd.openxmlformats-officedocument.wordprocessingml.document", "parent_id": "b19e7bb6fef7ba4f27ac6e47d2f7f88b", "header_footer_type": "primary", "page_content": "VANTIQ Integration Architecture Paul Butterworth June 5, 2019 Revised: June 12, 2019 Revision: 0.20 Draft I", "type": "Document"}}, {"id": null, "metadata": {"page_content": "VANTIQ Integration Architectures VANTIQ contains an extensive set of integration capabilities, supporting bot", "type": "Document"}}, {"id": null, "metadata": {"page_content": "driven application must be capable of obtaining the events from the external systems in an efficient and scala", "type": "Document"}}, {"id": null, "metadata": {"page_content": "AMQP Kafka The architecture of preconfigured sources is depicted in the following diagram illustrating the arc", "type": "Document"}}, {"id": null, "metadata": {"page_content": "Connectors for other external systems either integrate the appropriate Client API into the Enterprise Connect", "type": "Document"}}, {"id": null, "metadata": {"page_content": "Events may be delivered to instances of the VANTIQ platform running in the cloud, on premise or at the netw", "type": "Document"}}, {"id": null, "metadata": {"page_content": "A question that always arises is what kind of message rates a source can support. Unfortunately, the answer I", "type": "Document"}}
```

Document Loaders cont.

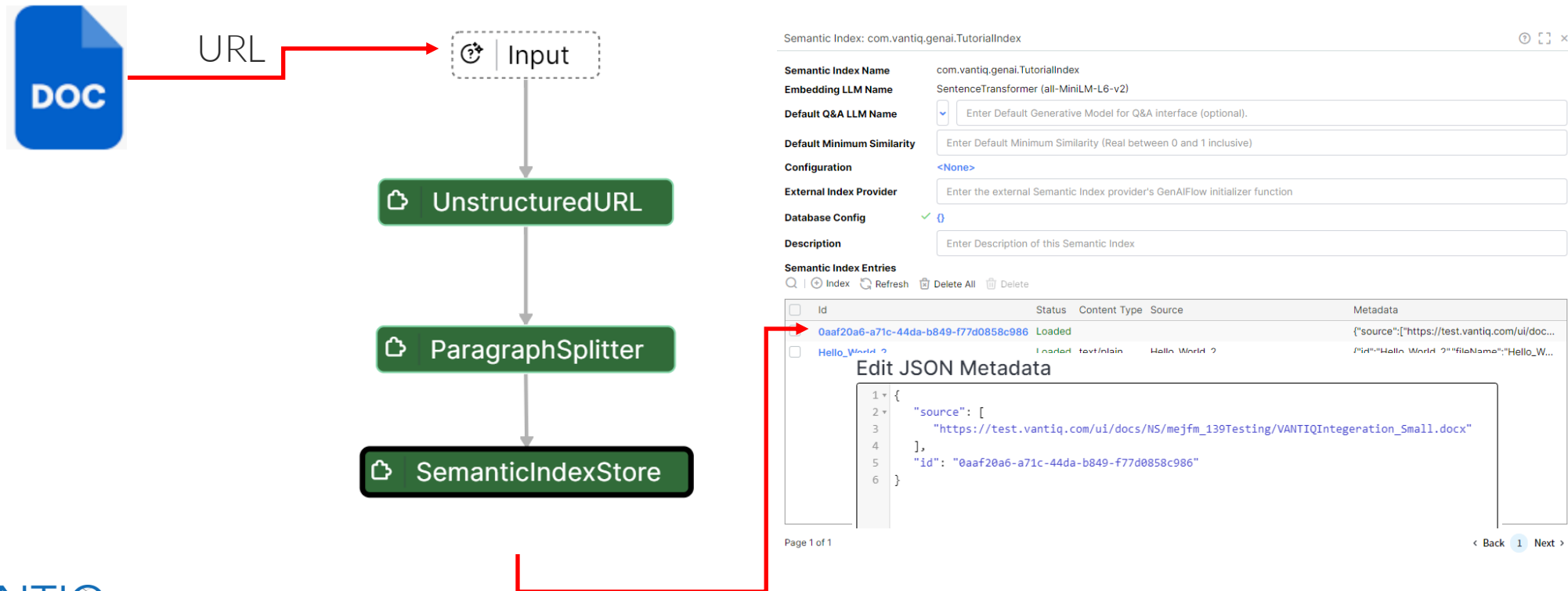
- ConversationMemory - converts a conversation, via the conversation ID, into a series of Documents

Document Transformers & Semantic Store

- The various Document Transformers accept a String, Array of Strings or Documents and split them into Chunks
 - HTMLSplitter
 - MarkdownSplitter - Splits a Markdown file by headers
 - ParagraphSplitter
 - CodeSplitter - define the language e.g. Java/C#/Python etc
- Each Splitter tries to keep related elements together within the given chunk constraints
- SemanticIndexStore can be used within a GenAI Flow to store a document(s) in a Semantic Index

Combining together

- The following example combines the UnstructuredURL, Paragraph Splitter and the Semantic Store

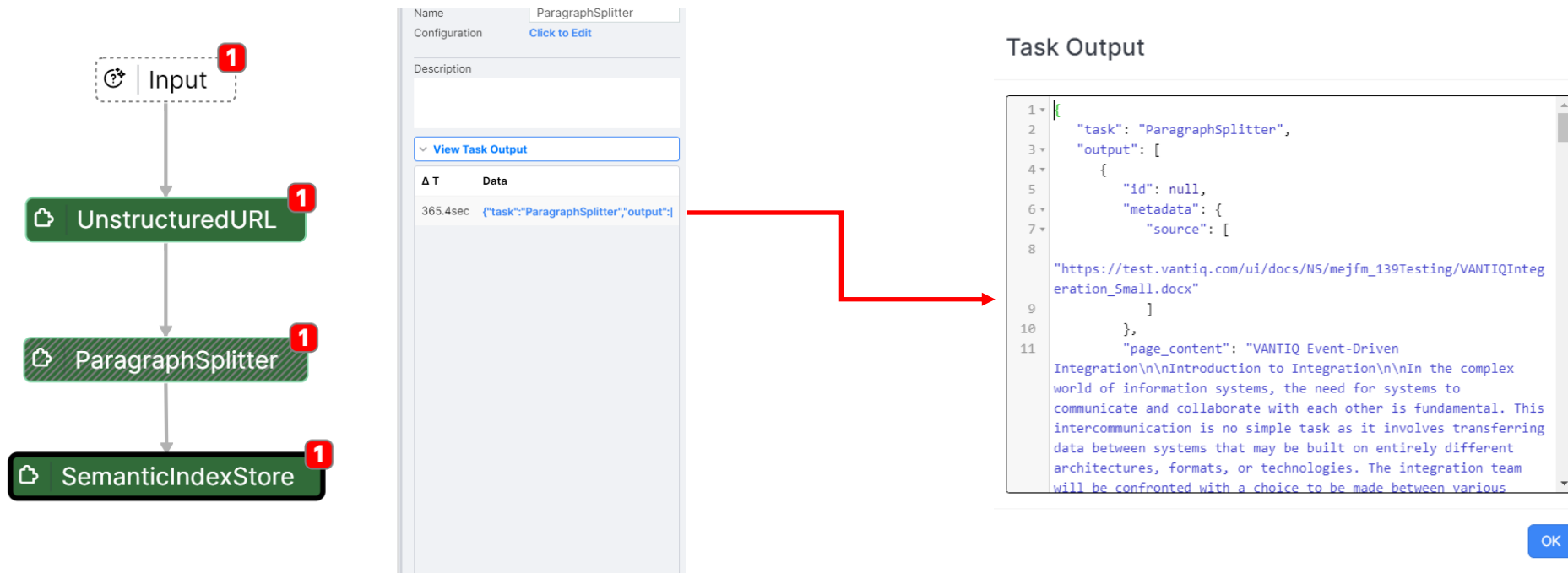




Other GenAI Builder & LLM Enhancements

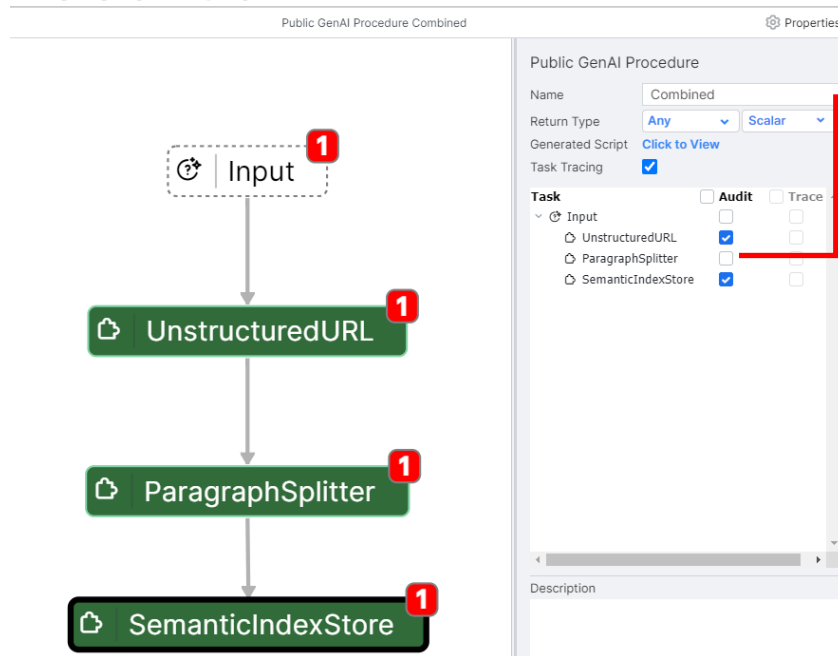
Badging and Tracing

- Badging - As per Visual Event Handlers
- View Task Output - See results from each GenAI Task, similar to View Task Events in Visual Event Handlers



Auditing

- Auditing within a GenAI Flow allows you to capture the results from any or all tasks in a GenAIFlow and record them as Audit records



Audit Query Results: Audit records after 08-27 10:59 with entity= '/system.genaiflows/com.vantiqse.DocIngestTest:Combined', operation= 'execu' [?] [] x

Q [] | Refresh

Timestamp	Entity	Operation	Message
2024-08-27 11:58:17.079	/system.genaiflows/com.vantiqse.DocIngestTest:Combined	execute	Executed the task SemanticIndexStore of GenAI Flow com.vantiqse.DocIngestTe...
2024-08-27 11:58:15.715	/system.genaiflows/com.vantiqse.DocIngestTest:Combined	execute	Executed the task UnstructuredURL of GenAI Flow com.vantiqse.DocIngestTest...

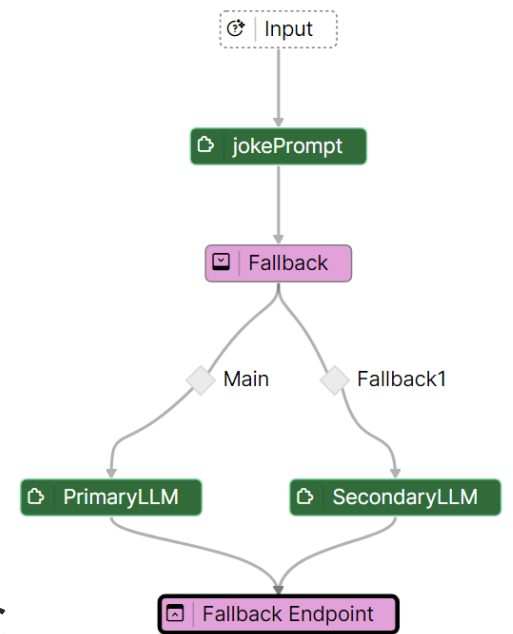
Audit Record

```
1 {
2   "_id": "66cdb1475a7fad305dcdc356",
3   "entity":
4     "/system.genaiflows/com.vantiqse.DocIngestTest:Combined",
5   "op": "execute",
6   "message": "Executed the task UnstructuredURL of GenAI Flow
7     com.vantiqse.DocIngestTest:Combined.",
8   "ars_properties": {
9     "task": "UnstructuredURL",
10    "output": {
11      {
12        "id": null,
13        "metadata": {↔},
14        "page_content": "VANTIQ Event-Driven
15      Integration\n\nIntroduction to Integration\n\nIn the complex
16      world of information systems, the need for systems to communicate
17      and collaborate with each other is fundamental. This
18      intercommunication is no simple task as it involves transferring
19      data between systems that may be built on entirely different
20      architectures, formats, or technologies. The integration team
```

OK

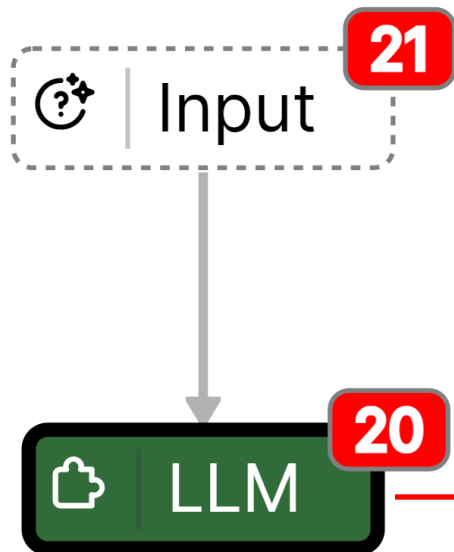
New GenAI Components

- **Fallback** - If the primary task fails, execute a fallback task. There can be multiple fallback tasks, and they are tried in order.
- **Tool** - GenAI Flow equivalent of Function calling (which has now been renamed to Tool) capabilities in the SubmitPrompt activity pattern.
 - Provide a list of Procedures or Services
 - Define the execution mode and authorizer
 - Also support custom Tools via code block
 - Only support Function enabled LLM's such as GPT 3.5 etc

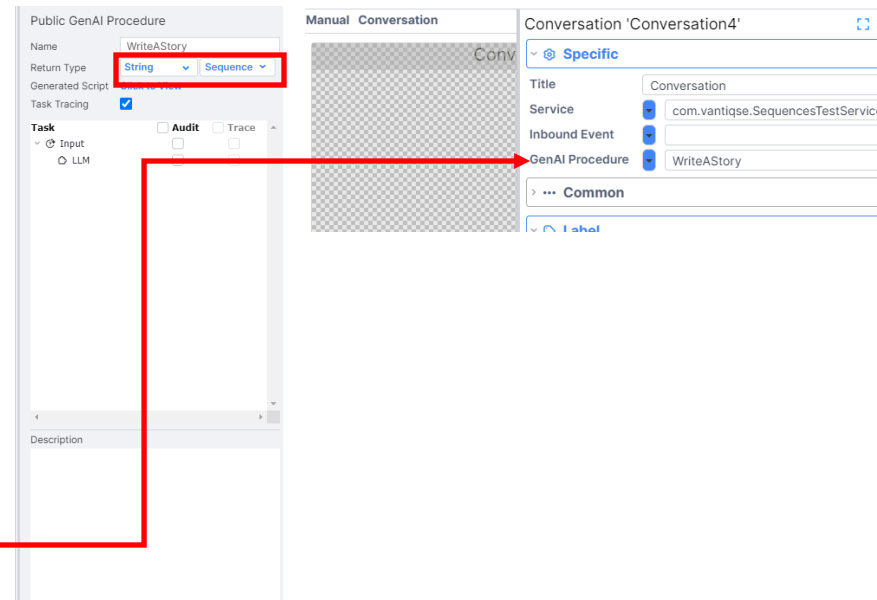


Streaming Support

- GenAI Flows now support streaming
 - Return type must be set to 'Sequence'
 - Allows you to get back partial results from the LLM



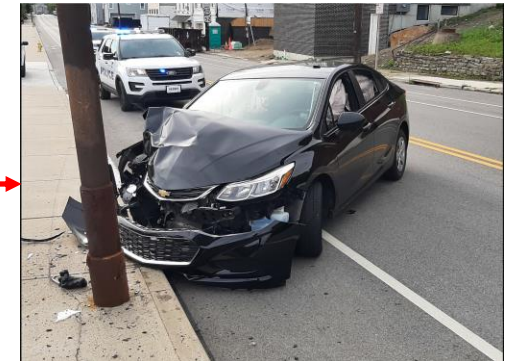
VANTIQ



Support for Multi-Modal models via Chat Message Builder

- ChatMessage.buildHumanMessage now supports image_url
- For instance, passing an image and question to a Multi Model LLM - below uses GPT-4o

```
package com.vantiqse
stateless PROCEDURE SequencesTestService.MultiModelQuestion()
var chat = []
var msg = io.vantiq.ai.ChatMessage.buildHumanMessage([
  {type: "text", text: "tell me about this picture"},
  {type: "image_url", image_url: {url: "https://test.vantiq.com/ui/docs/NS/mejfm_139Testing/car%20crash%201.jpg"}},
])
push(chat, msg)
var res = SequencesTestService.chatMsgFlow(chat)
```



Execution of 'com.vantiqse.SequencesTestService.MultiModelQuestion' complete

"The picture shows a black Chevrolet car that has crashed into a pole on the side of a road. The front of the car is significantly damaged, with the hood crumpled and the front bumper detached. The airbags inside the car appear to have deployed, indicating the impact was substantial. \n\nIn the background, there are two police vehicles with their lights on, suggesting that law enforcement is on the scene to manage the situation. The road appears to be in an urban area, with buildings and a sidewalk visible. Debris from the crash is scattered around the front of the car and the base of the pole. \n\nThe scene suggests a recent accident, and the presence of police indicates that the incident is being handled by authorities."

External Semantic Index Support

- Possible to integrate external semantic indexes into Vantiq - such as Elastic Search
- External Semantic Index then operates like the existing Semantic Index
- Required to extend GenAI Flow Service Connector and add additional functions that return Vector Stores
 - Contact support/eng
 - Elasticsearch_vector_store has been provided already

Semantic Index: New Semantic Index

Discard Save ? [] x

Semantic Index Name * ✓ MyExternalIndex

Package ✓ com.vantiq.test

Embedding LLM Name * ✓ com.vantiq.test.embedding

Default Q&A LLM Name ✓ Enter Default Generative Model for Q&A interface (optional).

Default Minimum Similarity Enter Default Minimum Similarity (Real between 0 and 1 inclusive)

Configuration <None>

External Index Provider ✓ elasticsearch_vector_store

External Index AnswerQuestion Procedure Enter the GenAI procedure to handle Q&A for this Semantic Index

Database Config ✓

Description Enter Description of this Semantic Index

```
{
  "host": "elastic.co.cloud.ep",
  "port": 443,
  "user": "elastic",
  "password": "@secrets(elasticsearch_password)"
}
```

Semantic Index and Qdrant changes

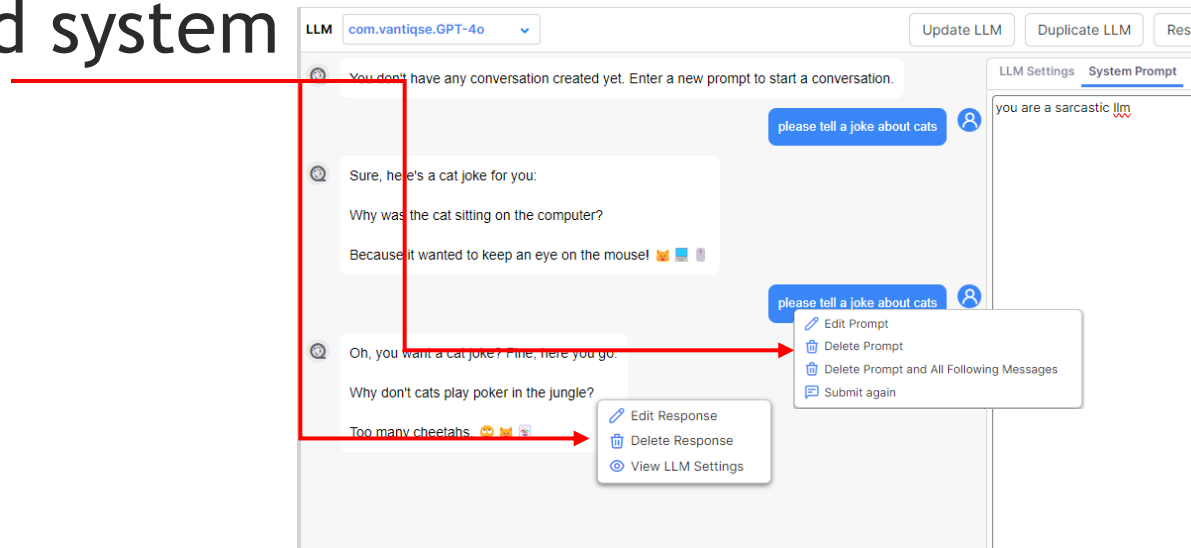
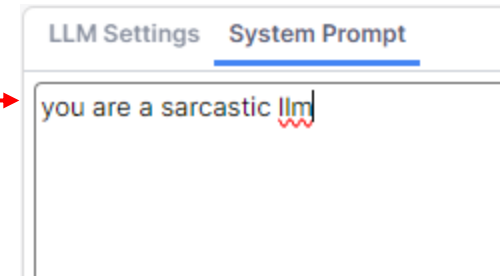
- Enhance support for Qdrant filters on the -
`SemanticSearch.similaritySearch(..., metadataFilter Object, ...)`
 - Explicit values e.g. `{"key1": "val1", "key2": "val2"}`
 - Or use the Qdrant filter format e.g. `{"must": [{"key": "key1", "match": {"value": "val1"}}, {"key": "key2", "match": {"value": "val2"}}]}`
- Semantic Index size is now unlimited - no longer restricted to MongoDB collection size of 16MB

The background features a complex geometric pattern of overlapping circles and intersecting lines in light blue and grey tones. A solid dark blue horizontal band spans the middle of the image, serving as a backdrop for the title text.

LLM Playground

LLM Playground

- Support for Conversation level System Prompts. The System Prompt is now defined own a tab next to the LLM settings.
- Support editing and resubmitting prompts - both user and system

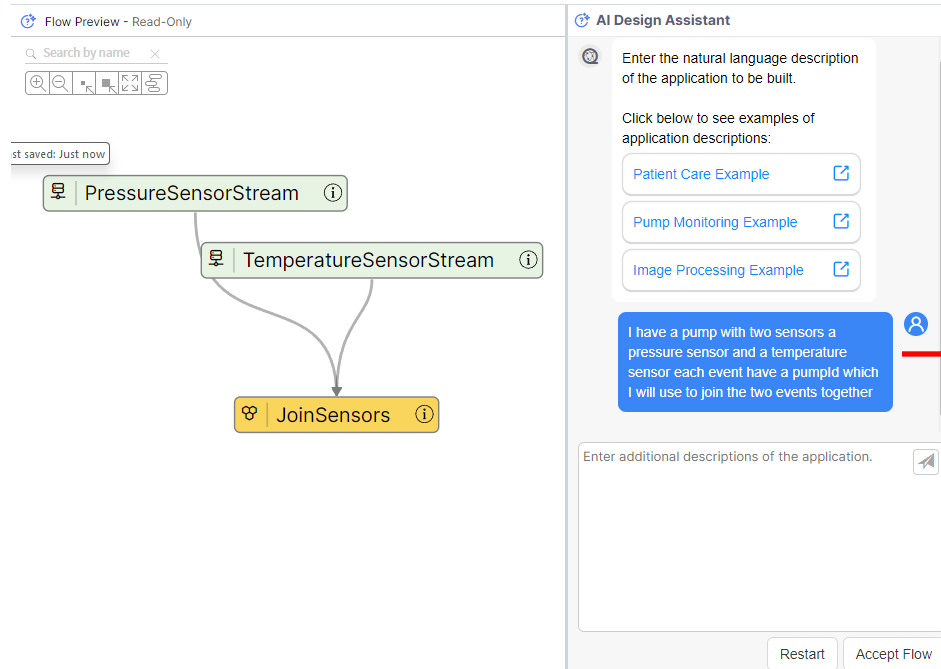




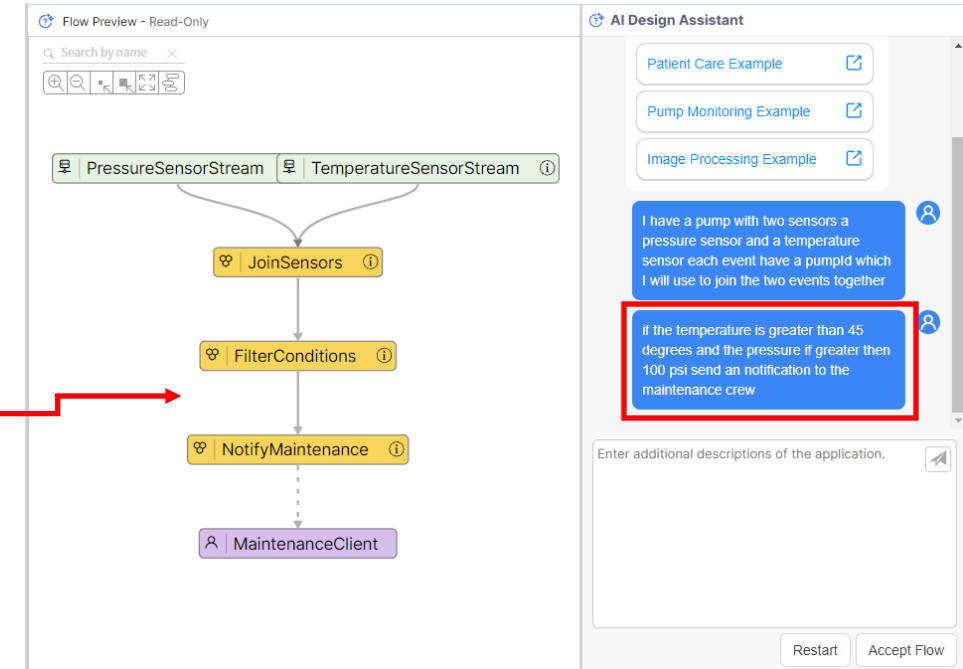
Design Model - AI Design Assistant Enhancements

AI Design Assistant Conversational Interface

- The chat interface now allows you to add to the flow by submitting additional messages



Initial msg builds the start of the flow. Follow-up msgs adds new functionality to the flow.

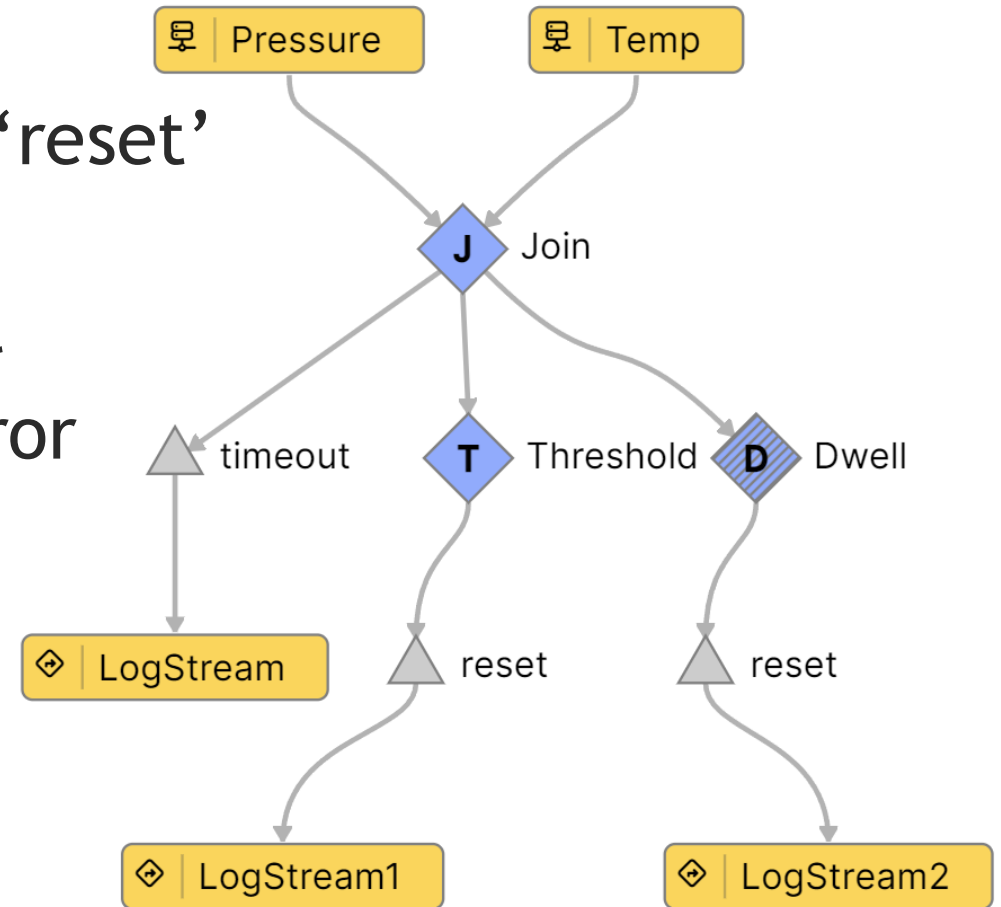




Server Enhancements

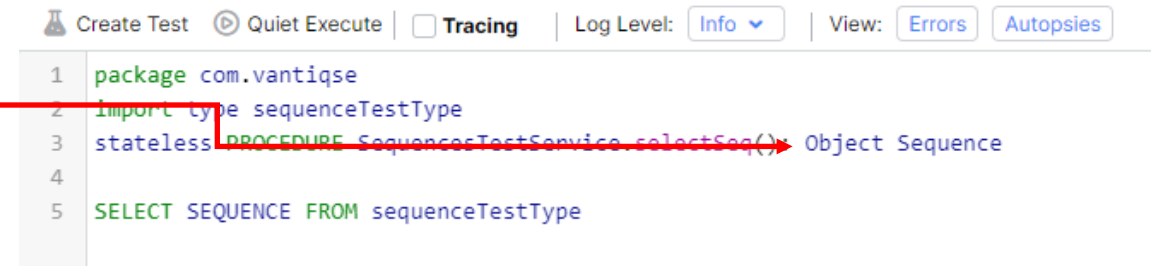
Activity Pattern Enhancements

- Dwell and Threshold now generate a 'reset' event
- Join now generates a 'timeout' event
- triggeringEvent property on the onError Event

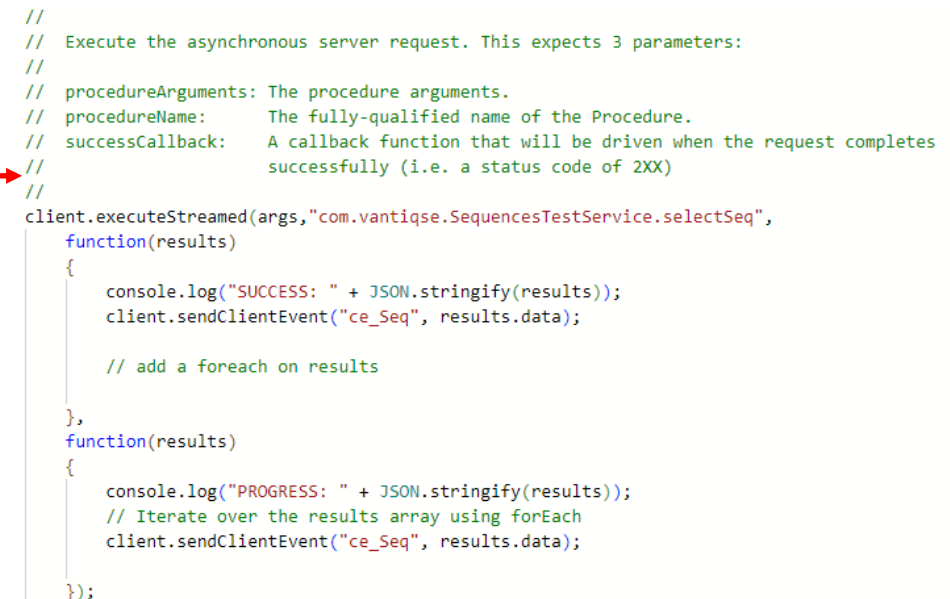


VAIL Procedures support Sequence return type

- VAIL procedures support a Sequence return type
 - This means that the results will be 'streamed' from the procedure
- From the ClientBuilder use the executeStream function



```
Create Test | Quiet Execute | ☐ Tracing | Log Level: Info | View: Errors Autopsies
1 package com.vantique
2 import type sequenceTestType
3 stateless PROCEDURE SequenceTestService.selectSeq() Object Sequence
4
5 SELECT SEQUENCE FROM sequenceTestType
```



```
//
// Execute the asynchronous server request. This expects 3 parameters:
//
// procedureArguments: The procedure arguments.
// procedureName:      The fully-qualified name of the Procedure.
// successCallback:    A callback function that will be driven when the request completes
//                    successfully (i.e. a status code of 2XX)
//
client.executeStreamed(args, "com.vantique.SequencesTestService.selectSeq",
    function(results)
    {
        console.log("SUCCESS: " + JSON.stringify(results));
        client.sendClientEvent("ce_Seq", results.data);

        // add a foreach on results
    },
    function(results)
    {
        console.log("PROGRESS: " + JSON.stringify(results));
        // Iterate over the results array using forEach
        client.sendClientEvent("ce_Seq", results.data);
    }
);
```



ClientBuilder Enhancements

Data Stream Filtering

- 1.39 added the ability to filter events at a WebSocket level
- DataStreams now support Filter Objects on Service/Topic/Source datastreams

```
{
  "op": "subscribe",
  "resourceName": "events",
  "resourceId": "/types/system.sources/insert",
  "parameters": {
    "requestId": "SubscriptionXYZ",
    "eventFilter": {
      "type": "REMOTE"
    }
  }
}
```

Create New Data Stream

On Service Event

Data Stream Name * ✓ se_ClientTestOutbound

Event Filtering [Click to create an event filter object](#)

Service and Event * ✓  com.vantiqse.ClientTestService

OutTemp

Group By (Optional) Enter property

On Data Arrived
<None>

☐ Pause Data Stream at Client start ⓘ

Edit Event Filter Object

Setting this object causes incoming events to be filtered. Only those events which contain properties that match the specified values will be delivered.

1 {"type": "REMOTE"}

Conversation Widget Enhancements

- Conversation Widget supports Service Events and GenAI Procedures now
- Support for streaming output for GenAI procedures that return a 'Sequence'
- Conversation Widget will participate in the Default conversation within a collaboration

‘Stream’ client functions

- Two new function added to the Client helper to support ‘Streaming’ procedures (Vail or GenAI)
 - `executeStreamed(procedureArguments:any, procedureName:string, succeed:Function, progress:Function):void`
 - `executeStreamedPublic(namespace:string, procedureArguments:any, procedureName:string, succeed:Function, progress:Function):void`
 - The ‘progress’ function is called when every ‘chunk’ arrives. The ‘succeed’ function is called when the last ‘chunk’ arrives
 - ‘progress’ and ‘succeed’ results includes
 - `data` - The cumulative data received (so far) as an array
 - `rawData` - The cumulative raw data received for far as a string (this might be parseable as a complete array, but it might be truncated such that the parse would result in an error)
 - `isComplete` - a boolean set to “true” when no more callbacks are expected
 - `chunksReceived` - (Progress Only) Total “chunks” received so far
 - `totalRowsReceived` - (Progress Only) Total rows received so far
 - `newRowsReceived` - (Progress Only) New rows received with this chunk
 - `firstNewRowIndex` - (Progress Only) The index of the first new row in this chunk

'Stream' client function example

```
11 // Set the Procedure arguments by name. (You may also specify 'args' as an array where the
12 // parameters are given in the same order as in the Procedure definition (e.g. 'args = [10,20];').
13 // 'args' must not be null.
14 //
15 var args = {
16 };
17
18 //
19 // Execute the asynchronous server request. This expects 3 parameters:
20 //
21 // procedureArguments: The procedure arguments.
22 // procedureName:      The fully-qualified name of the Procedure.
23 // successCallback:    A callback function that will be driven when the request completes
24 //                     successfully (i.e. a status code of 2XX)
25 //
26 client.executeStreamed(args,"com.vantiqse.SequencesTestService.selectSeq",
27     function(results)
28     {
29         console.log("SUCCESS: " + JSON.stringify(results));
30         client.sendClientEvent("ce_Seq", results.data);
31
32         // add a foreach on results
33
34     },
35     function(results)
36     {
37         console.log("PROGRESS: " + JSON.stringify(results));
38         // Iterate over the results array using forEach
39         client.sendClientEvent("ce_Seq", results.data);
40
41     });
42
```

Launch Client into a collaboration context

- The Launch menu has been enhanced to support launching into a collaboration context

The image shows a screenshot of the VantIQ Launch menu and a dialog box. The menu is divided into two sections. The top section contains four items: 'Run currently saved Client in Client Launcher (RTC)', 'Copy Client Launcher URL to Clipboard', 'Run currently saved Client in Mobile Launcher (MPI)', and 'Copy Mobile Launcher URL to Clipboard'. The bottom section contains three items: 'Run Client under Collaboration Context', 'Run currently saved Client in Mobile Launcher (MPI) under Collaboration', and 'Copy Mobile Launcher URL to Clipboard under Collaboration'. A red arrow points from the 'Run Client under Collaboration Context' item to a dialog box titled 'Run the Client under an active Collaboration'. The dialog box has a 'Collaboration ID' dropdown menu with a placeholder text 'Enter a valid active Collaboration Id'. Below the dropdown, two collaboration IDs are listed: 'com.vantiq.MyService.Launch (1b3c9200-1f6f-11ef-9268-d60834ff2cab)' and 'com.vantiq.MyService.Launch (bf3e3e80-2181-11ef-b773-360d774ccf0b)'. A text label 'Select the Collaboration to use' with a red arrow points to the dropdown menu.

Select the Collaboration to use

Run the Client under an active Collaboration

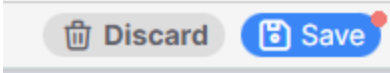
Collaboration ID

com.vantiq.MyService.Launch (1b3c9200-1f6f-11ef-9268-d60834ff2cab)
com.vantiq.MyService.Launch (bf3e3e80-2181-11ef-b773-360d774ccf0b)



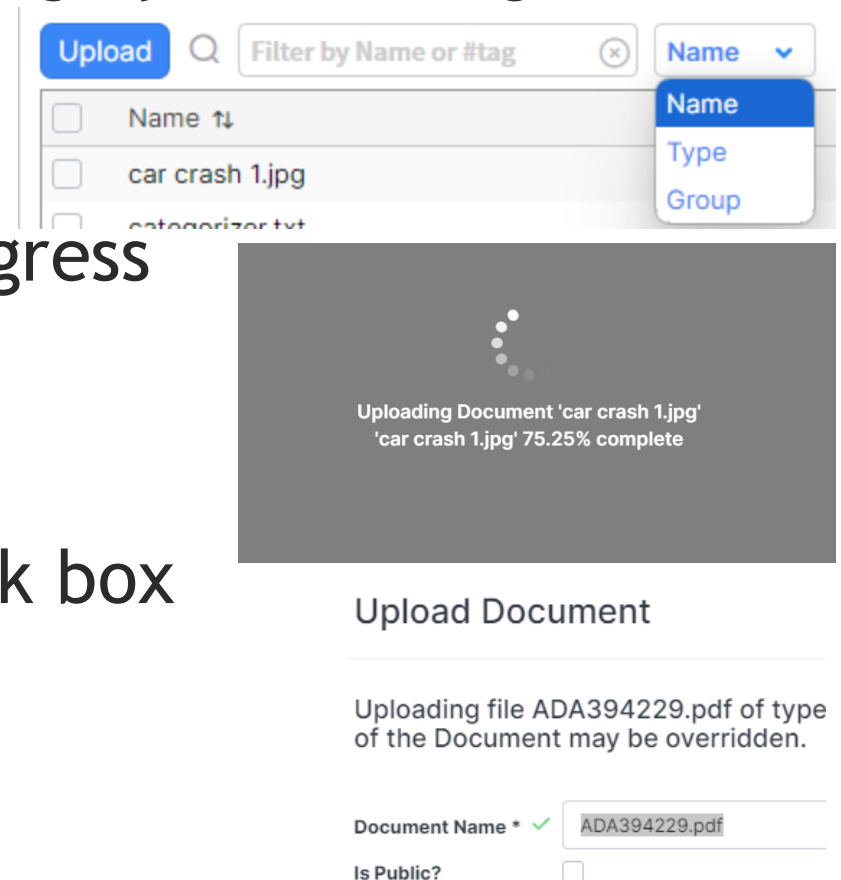
IDE Enhancements

General IDE Enhancement

- Auth Token timeout and count-down timer
 - IDE shows a countdown timer or a message indicating that the token has expired 
- Discard - 
 - Various builders and editor now support 'discard' - ServiceBuilder, ClientBuilder, LLM, Semantic Index, etc.
 - Discard will discard any changes and revert the editor/builder back to its previous saved state

General IDE Enhancement Documents

- Document - Search now supports filtering by name, tag, or type
- Uploading Documents now shows % progress
- Upload document has a 'Is Public' check box



The screenshot displays the 'Upload Document' section of the VANTIQ IDE. At the top, there is an 'Upload' button, a search icon, and a search bar labeled 'Filter by Name or #tag'. To the right of the search bar is a dropdown menu currently set to 'Name', with options for 'Name', 'Type', and 'Group'. Below the search bar is a table listing documents with checkboxes and names: 'Name ↑↓', 'car crash 1.jpg', and 'categorizer.txt'. A large grey box in the center shows a progress bar and the text 'Uploading Document 'car crash 1.jpg' 'car crash 1.jpg' 75.25% complete'. Below this, the 'Upload Document' title is followed by a message: 'Uploading file ADA394229.pdf of type of the Document may be overridden.' At the bottom, there is a 'Document Name' field with a green checkmark and the text 'ADA394229.pdf', and an 'Is Public?' checkbox which is currently unchecked.

Semantic Index Enhancements

- Uploading Files - now support selecting multiple files. Files will be 'zipped' and uploaded

The diagram illustrates the file upload process flow. It starts with a 'Confirm' dialog box asking if the user wants to add 3 selected files as a single index entry or separately. Two paths are shown: a red path for 'Add Single Entry' and a blue path for 'Add 3 Separate Entries'. The red path leads to a file list table where the files are listed as individual entries. The blue path leads to a file list table where the files are listed as a single zipped entry.

Confirm Dialog:

Confirm

You selected 3 Files; do you want to add them all in a single index entry or separately?

File List (Red Path - Single Entry):

Filename *	Id
3 Files: Domain Integration and Vantiq - military - v1.4.pdf,VANTIQ Eve...	Enter unique ID

File List (Blue Path - Multiple Entries):

Filename *	Id
tempfile.zip	

File List Table:

☐	Id	Status	Content Type	Source	Metadata
☐	2da47120-652d-11ef-8fad-fe5d5c1b8f83	Loaded	application/zip	tempfile.zip	{"global":{"id":"2da47120-652d-11ef-8fad...
☐	Domain Integration and Vantiq - mili...	Loaded	application/pdf	Domain Integration and Vantiq - military...	{"id":"Domain Integration and Vantiq - mili...
☐	VANTIQ Event Driven Integration - v1...	Loaded	application/pdf	VANTIQ Event Driven Integration - v1.1.p...	{"id":"VANTIQ Event Driven Integration - ...
☐	VANTIQ Platform Security Architectur...	Loaded	application/pdf	VANTIQ Platform Security Architecture ...	{"id":"VANTIQ Platform Security Architect...



AWS Marketplace

Vantiq Developer Edition now available on the AWS Marketplace



Vantiq Developer Edition

By: [VantIQ](#) Latest Version: 1.38.13_1

The patented Vantiq platform is built to handle a new era of enterprise applications where sensors, AI and business systems all work together in cohesive synergy and in real-time to

[Show more](#)

Linux/Unix

[Continue to Subscribe](#)

[Save to List](#)

Typical Total Price
\$0.192/hr
Total pricing per instance for services hosted on m6i.xlarge in US East (N. Virginia). [View Details](#)

[Overview](#)[Pricing](#)[Usage](#)[Support](#)[Reviews](#)

Product Overview

The VANTIQ Developer Edition is a free version of the VANTIQ platform available on the AWS Marketplace. It provides full access to the core features: Experience real-time data processing; event-driven architecture; integration of Generative AI (AWS Bedrock and SageMaker) into real-time applications; integrate RAG using the built-in Vector Database; visual development environment -- All within the AWS cloud infrastructure.

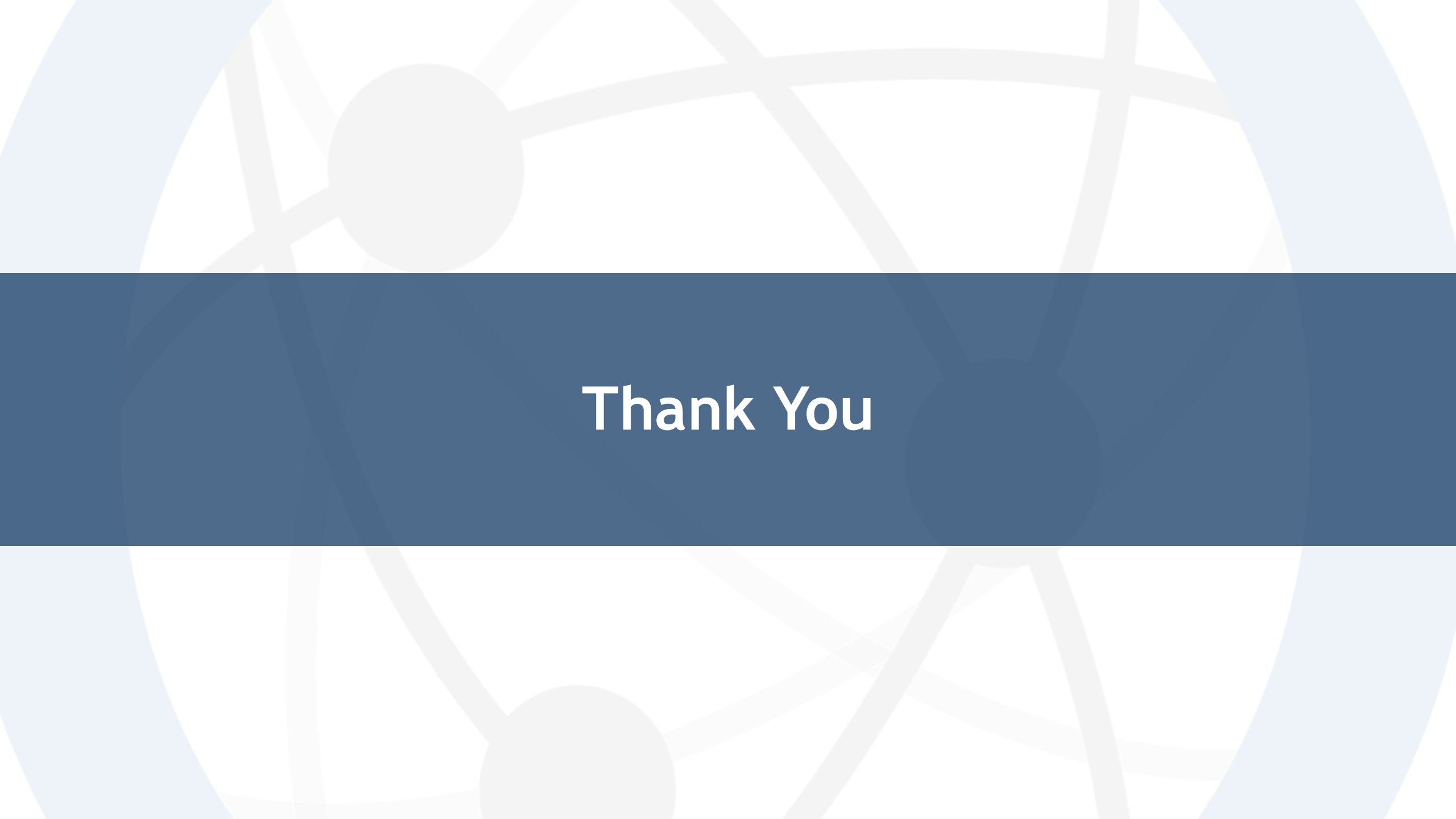
The VANTIQ Developer Edition on AWS Marketplace is a golden opportunity for developers to dive into the world of real-time, event-driven application development. With zero costs and a plethora of resources at their fingertips, developers can innovate, learn, and prepare for the demands of the modern digital world. Whether you're a seasoned developer or just starting out, the VANTIQ Developer Edition is your gateway to the future of application development.

Version	1.38.13_1
By	VantIQ
Video	See Product Video
Categories	Application Development Generative AI Analytics
Operating System	Linux/Unix, Amazon Linux 5.10
Delivery Methods	Amazon Machine Image

Highlights

- Vantiq excels in handling real-time data streams and event processing. It allows developers to build applications that can react instantly to events, and complex scenarios. Uses a service-based approach, that supports event-driven asynchronous interfaces and synchronous interfaces. Support for in-memory state for managing high-performance complex event interactions. The Low-code IDE allows developers to design, and develop with minimal coding (90% graphical), speeding up the development process.
- Vantiq automatically prompts Generative AI to amplify contextual insights in real-time situations. Enables dynamic communication between Machine-to-Machine and Machine-to-Human interactions and orchestrates across multiple LLMs with patented voting/consensus algorithms. Empowers real-time decision-making and business automation.
- Integrate and orchestrate AI (Generative AI, Computer Vision, Machine Learning), IoT Data, and existing systems. Out-of-the-box support for AWS Bedrock and SageMaker LLMs as well as OpenAI and other LLMs. Inbuilt support for RAG using a Vector Database. Supports 160+ integrations and provides SDK and API to allow developers to provide custom integrations. Ease of integration with external Rest-based services.





Thank You