

VANTIQ

VANTIQ 1.41 Update

VANTIQ Global Field & Partner Enablement

Agenda

- Overview of New Features and Enhancements
- GenAI Builder - Document Compressors
- GenAI Builder - General Enhancements
- Client Builder - AI Assistant
- LLM Playground Enhancements
- Visual Event Handler Enhancements
- IDE Enhancements
- Server Enhancements
- Video Source
- React Native Library

Overview of New Features and Enhancements

Integration Enhancements - GenAI Builder Document Compressors

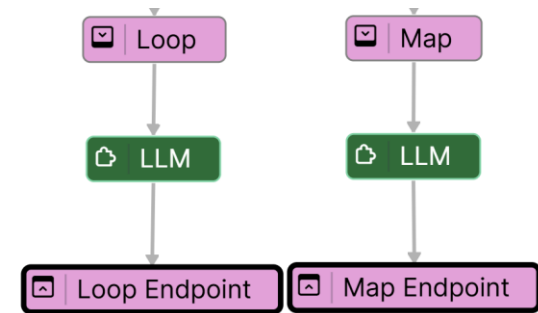
Increased integration capabilities

- Document Compressor Overview
 - Reranking and document compressors work together to significantly improve the relevance and quality of generated responses by carefully selecting the most pertinent information from a retrieved set of documents in a RAG pipeline
 - Extract Relevant Document and Filter Relevant Content both use an LLM to extract or filter documents
 - CohereRerank and LikewiseRerank rerank the documents using a reranking algorithm or LLM

Usability Enhancements - GenAI enhancements

Increased developer productivity

- GenAI Builder - Additional Components
 - Loop, Map and Transform
 - Call VAIL Procedures (not Tools)
 - SemanticIndexWithCompression
 - Perform a semantic index search and applies one or more document compressors or transformers against the array of documents
 - ReduceDocument Pattern summaries a series of documents
- Semantic Index - Content ingestion can now be implemented via a GenAI procedure

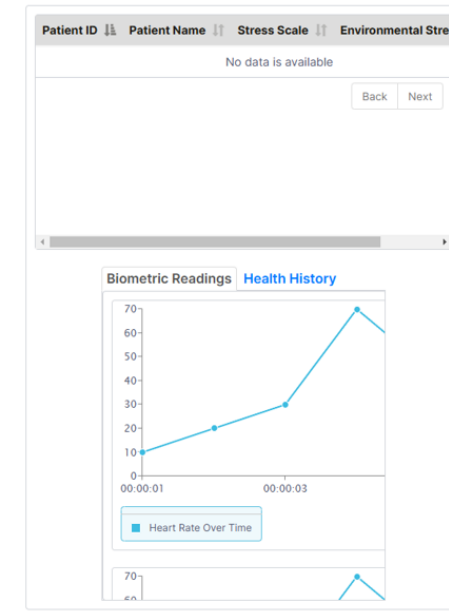


Semantic Index: com.vantiq.TestIndex	
Semantic Index Name	com.vantiq.TestIndex
Embedding LLM Name	com.vantiq.embedding (all-MiniLM-L12-v2)
Default Q&A LLM Name	☒ com.vantiq.gpt
Default Minimum Similarity	Enter Default Minimum Similarity (Real between 0 and 1 inclusive)
Configuration	<None>
Ingest GenAI Procedure	Enter the GenAI procedure to handle ingestion for this Semantic Index (optional)

Productivity Enhancements - Client Builder AI Assistant

Increased developer productivity

- Client Builder AI Assistant
 - Allows the developer to create Client Builder clients using an AI Assistant
 - The developer provides a description, and the AI Assistant will generate the Client
 - Conversational interaction



AI Client Assistant



Enter the natural language description of the page to be built.

Click below to see examples of page descriptions:

[Patient Care Example](#)



[Pump Monitoring Example](#)



[Image Processing Example](#)



Produce a display that lists each patient's value on the stress scale highlighting the patients that are exceeding the stress limits. Include in the list, the environmental stress score for each patient's location. Allow the user to select a patient and drill down into the details of that patient's biometric readings and their relevant health history which can be obtained from their medical records in their EHR system.



Integration Enhancements - Video Source and React Native Client SDK

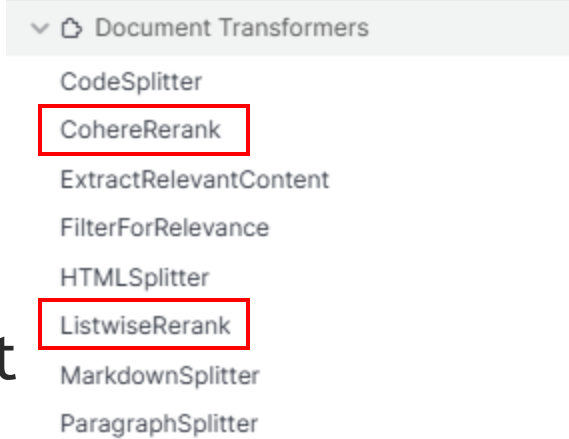
Increased integration capabilities

- Video Source and Video Assistant allow developers to connect to video streams and ingest individual frames as images
 - Images can be stored temporarily in memory or on disk in `system.documents`
- React Native SDK
 - Allows developers who are building React Native Mobile Applications to easily integrate with the VantIQ server
 - Installed via NPM and supports both IOS and Android

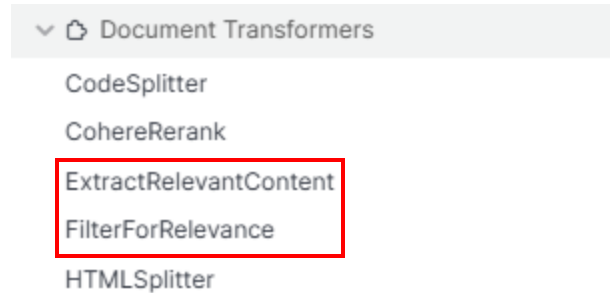
GenAI Builder Document Compressors

Document Reranking

- Two new reranking components
 - Both accept a query and one or more documents as input
- Cohere Rerank
 - Integrates a Cohere Reranking model - rerank-v3.5, rerank-English-v3.0 and rerank-multilingual-v3.0 and any that Cohere add later (string name)
 - Needs Cohere API key
- ListwiseRerank
 - Uses a language model to rerank a list of documents based on their relevance to a query.



Document Compressors



- Two document compressors
 - These allow you to either further filter the documents from a semantic search or compress/extract only the relevant information from the documents
- FilterForRelevance
 - Uses an LLM to decide which of the initially retrieved documents to filter out and which ones to return, without manipulating the document contents
- ExtractRelevantContent
 - ExtractRelevantContent will iterate over the documents and extract from them only the content that is relevant to the query

ReduceDocument - AI Pattern

- ReduceDocuments is an AI Pattern rather than a low-level component
 - It is a complex pattern as it will split documents that are larger than the token size specified
 - Uses an LLM and Prompt to reduce an array of documents into a single document
 - See <https://python.langchain.com/docs/tutorials/summarization/#map-reduce>

Uses the given prompt and LLM to reduce an array of documents to a single result.

Required Parameter	Value
reducePrompt (Union)	<null> <i>The prompt used to perform the reduce on the supplied documents.</i>
reduceLLM (LLM)	 <i>The LLM used to perform the reduce operation.</i>

Optional Parameter	Value
maxTokensPerBatch (Integer)	8196 <i>The maximum number of tokens permitted in a single reduce request. It is an error if any of the supplied documents exceeded this threshold.</i>
maxPartialReductions (Integer)	 <i>The maximum number of "partial" reductions allowed before the final result is achieved. If no value is provided then the reduction process will continue until done.</i>

Other GenAI Builder & LLM Enhancements

New GenAI Components

- Procedure - Execute a VAIL procedure from within a GenAI Flow, not the same as Tools

Execute the specified procedure using the parameters returned by the code block.

Required Parameter	Value
procedure (Union)	<null> <i>Specify the name of the procedure to execute.</i>

Optional Parameter	Value
parameters (Code Block)	<null> <i>Code that must return an Object (VAIL) or dictionary (python) expressing the procedure parameters, where key names are the parameter names and values are the parameter values.</i>

- Transform - Visual or programmatic transformation - similar to Visual Event Handler Transforms
 - Can access flowstate, see later

transformation Type: Visual Transformation

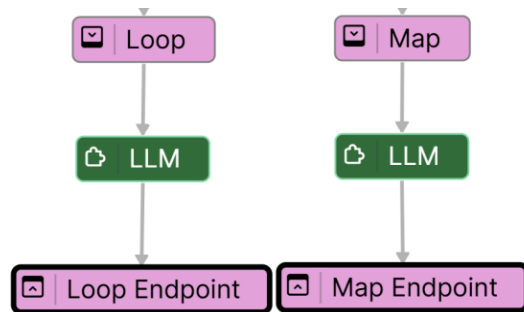
transformation: Projection

[Add a Transformation](#) [Create Schema](#) [Reset Transformation](#) [Remove All](#)

Outbound Property	Transformation Expression	Actions
-------------------	---------------------------	---------

New GenAI Components - Loop & Map

- **Loop** - Repeat the enclosed sub-flow while the specified condition evaluates to true. The condition is checked at the start of each iteration, and the loop terminates when the condition evaluates to false.



- **Map** - Take an array as an input and executes the sub-flow for each element of the array. Will execute each sub-flow concurrently

Loop on a sub-flow until the defined condition returns 'false' or the loop limit is reached.

Required Parameter	Value
condition (Code Block)	<null> Code that must return a boolean value. If the returned value is true, the loop keeps executing the sub-flow. If the returned value is false, the loop ends.

Optional Parameter	Value
loopLimit (Integer)	20 The maximum number of loop iterations allowed. If the loop condition is still 'true' when this limit is reached, the loop exits with an error.

Executes the contained sub-flow once for each item in the supplied array. Returns an array containing each of the resulting values.

Optional Parameter	Value
withIndex (Boolean)	☐ Whether or not to return the index of the input value used to produce each output value. When set to 'true', the return will be an Object with the 'index' and 'value' properties. Otherwise, only the value produced by the sub-flow is returned.
maxConcurrency (Integer)	 Maximum number of concurrent executions. Must be an Integer greater than or equal to 1. In general this should not be set unless sequential execution is required.

Flowstate - special memory management

- A new memory state for each GenAI Flow
 - Flowstate is represented as a VAIL Object or a Python dictionary
 - Is accessible from any CodeBlock

```
flowstate['count'] = flowstate.get('count', 0) + 1
return input
```
 - Is also accessible in other GenAI Components
 - Branch - accessible in the Branch Expressions
 - Loop - accessible in the condition code block
 - Procedure - accessible to the procedure name definition expression and the parameters code block
 - Transform - accessible to the transformation code block
 - Or any component that includes an expression or code block

GenAI Builder Enhancements

- Optional and Repeat are now in a new category ‘Configuration Processors’
- LLM AI Components have an optional output type schema property that will format the output according to the schema when using JSON output

Optional Parameter	Value
outputType (Enumerated)	Json Parse the response from the LLM and transform it based on the given type.
outputTypeSchema (Type)	com.vantiq.TestService.TestEvent Try to format the response object into a given schema. Only used when the outputType is "Json".

Ability to configure System Prompt on an LLM

- The LLM resource now has the ability to define the system prompt that will be used for all interactions with this LLM

LLM: New LLM

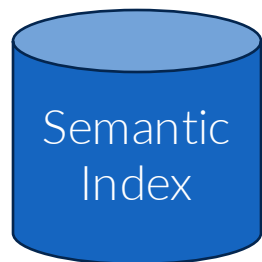
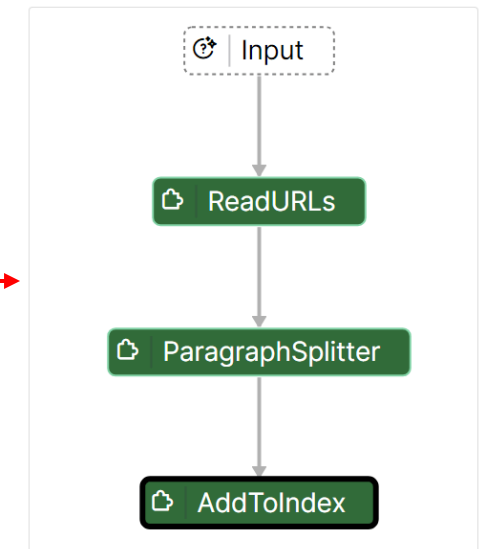
LLM Name *	✓	gpt
Package	✓	com.vantiq
Type *	✓	☐ Embedding ☒ Generative
Model Name *	✓	gpt-4o
Description		Enter Description of this LLM
Configuration		null
API Key Secret	✓	openai_key
System Prompt		

Semantic Index GenAI Flow Content Ingestion

- It is now possible to configure a GenAI procedure to override the default content ingestion process
- Normal 2 min execution limit does not apply now 2 hours

Semantic Index: com.vantiq.TestIndex

Semantic Index Name	com.vantiq.TestIndex
Embedding LLM Name	com.vantiq.embedding (all-MiniLM-L12-v2)
Default Q&A LLM Name	☒ com.vantiq.gpt
Default Minimum Similarity	Enter Default Minimum Similarity (Real between 0 and 1 inclusive)
Configuration	<None>
Ingest GenAI Procedure	☐ Enter the GenAI procedure to handle ingestion for this Semantic Index (optional)



The AddToIndex will be overridden and the resulting document will be stored in the index that called the procedure

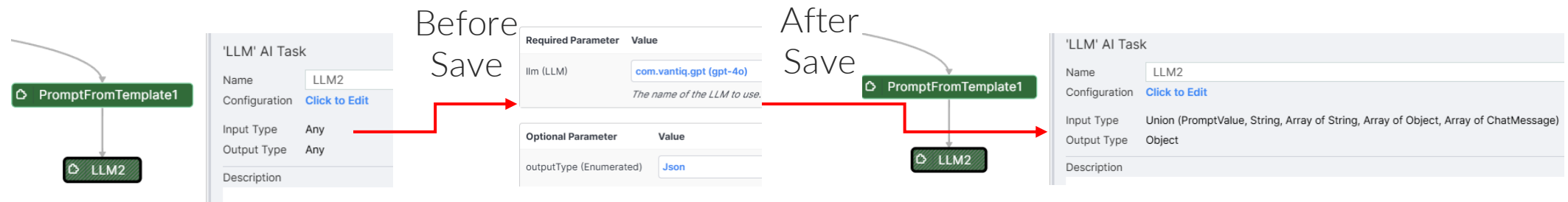
Chat Message Builder Image URL changes

- The ChatMessage Builders image url can now be a resource reference or a TempBlob (more on TempBlob's later)

```
var msg = io.vantiq.ai.ChatMessage.buildHumanMessage([
  {type: "text", text: prompt},
  {type: "image_url", image_url: {url: imageUrl}},
])
```

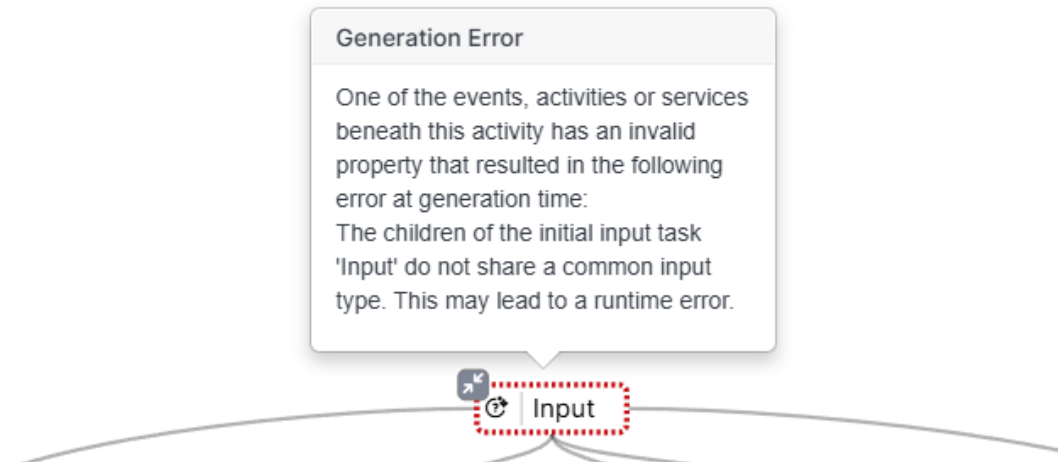
GenAI Builder Type hints and compilation errors

- GenAI Builder now provides type hints indicating what the tasks inputs and outputs are
 - Most components have a predefined list of known input and output types
 - Some components can supply a type such as the Transform component
 - After configuring a component and saving the types may change based on the configuration



GenAI Builder Type hints and compilation errors

- When compiling, the GenAI Builder will now show possible runtime issues based on flow definition





Client Builder AI Assistant

Client Builder AI Assistant

The screenshot shows the 'Client: TestAIAssistant' window. The top toolbar includes 'Discard', 'Save', and 'Launch' buttons. A red box highlights the 'AI Client Assistant' button. The main workspace displays a table with columns: Patient ID, Patient Name, Stress Scale, and Environmental Stress. Below the table is a line chart titled 'Biometric Readings' showing 'Heart Rate Over Time'. The chart has a y-axis from 0 to 70 and an x-axis with time markers 00:00:01 and 00:00:03. A legend indicates 'Heart Rate Over Time' with a blue line. The chart shows a line starting at 10, rising to 20, then 30, and finally 70. Below the chart is a small table with a single row showing a value of 70.

Client: TestAIAssistant

Discard Save Launch AI Client Assistant

Undo Redo Delete Cut Copy Paste Clipboard Launch

Add Edit

Filter by Name

Layout Buttons Data Display Data Input

Patient ID Patient Name Stress Scale Environmental Stress

No data is available

Back Next

Biometric Readings Health History

Heart Rate Over Time

Heart Rate

Launch AI Assistant

Generate Client from descriptions

AI Assistant is conversational – you can adjust what is generated

The screenshot shows the AI Assistant interface. It displays a table with columns: Patient ID, Patient Name, Stress Scale, and Environmental Stress. Below the table is a line chart titled 'Biometric Readings' showing 'Heart Rate Over Time'. The chart has a y-axis from 0 to 70 and an x-axis with time markers 00:00:01 and 00:00:03. A legend indicates 'Heart Rate Over Time' with a blue line. The chart shows a line starting at 10, rising to 20, then 30, and finally 70. Below the chart is a small table with a single row showing a value of 70. A red box highlights the 'remove Health History' button.

user interface.

remove Health History

The application will have a clean and structured user interface with the following sections:

1. **Patient Stress List**:

- This section will display a table listing each patient's stress scale value. Patients exceeding the stress limits will be highlighted in red.
- The table will also include the environmental stress score for each patient's location.
- Sample Data:
- Columns: Patient ID, Patient Name, Stress Scale, Environmental Stress Score
- Example Rows:
- Patient ID: 001, Patient Name: John Doe, Stress Scale: 85 (highlighted), Environmental Stress Score: 70
- Patient ID: 002, Patient Name: Jane

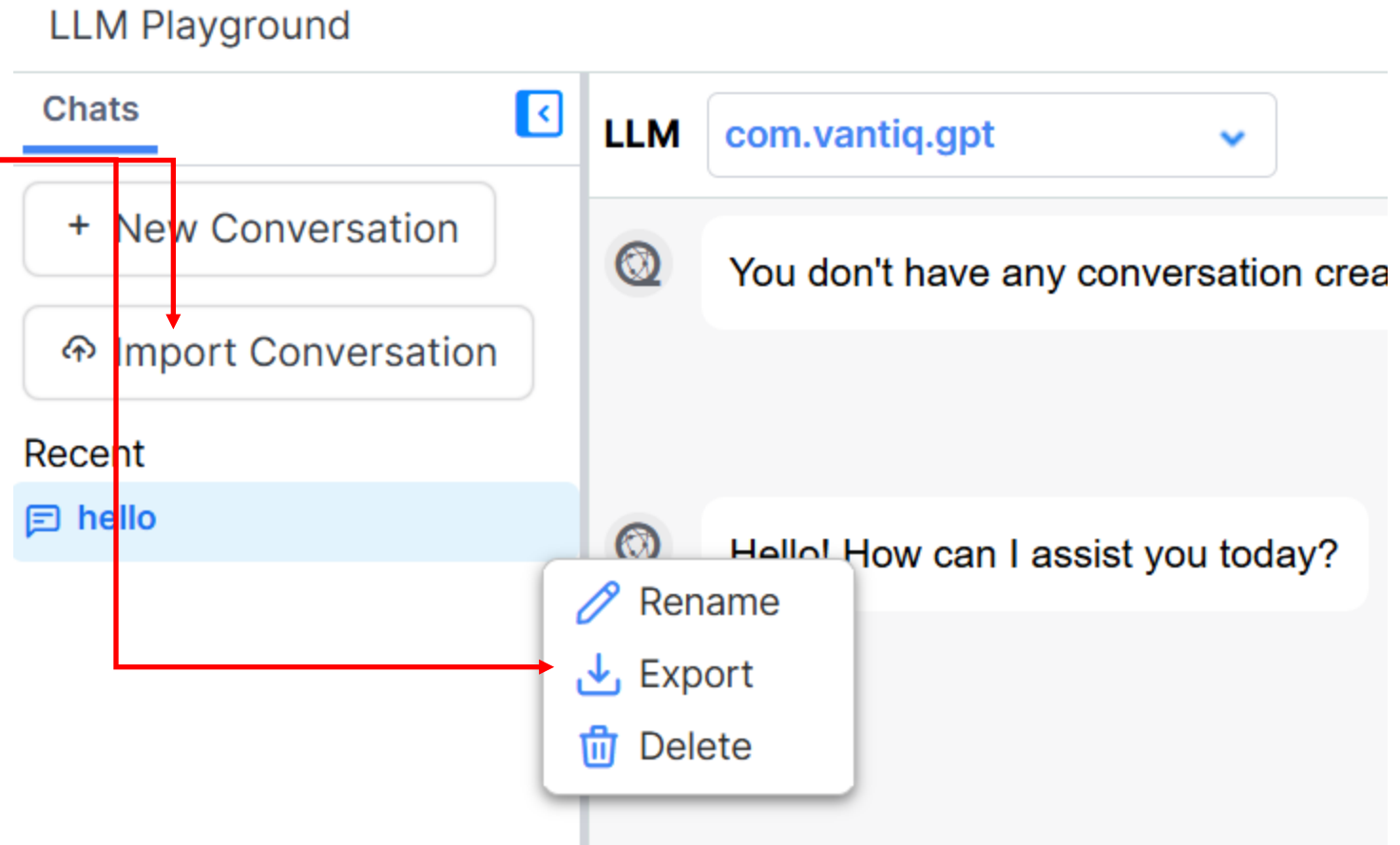
Enter the natural language description to modify the generated page.

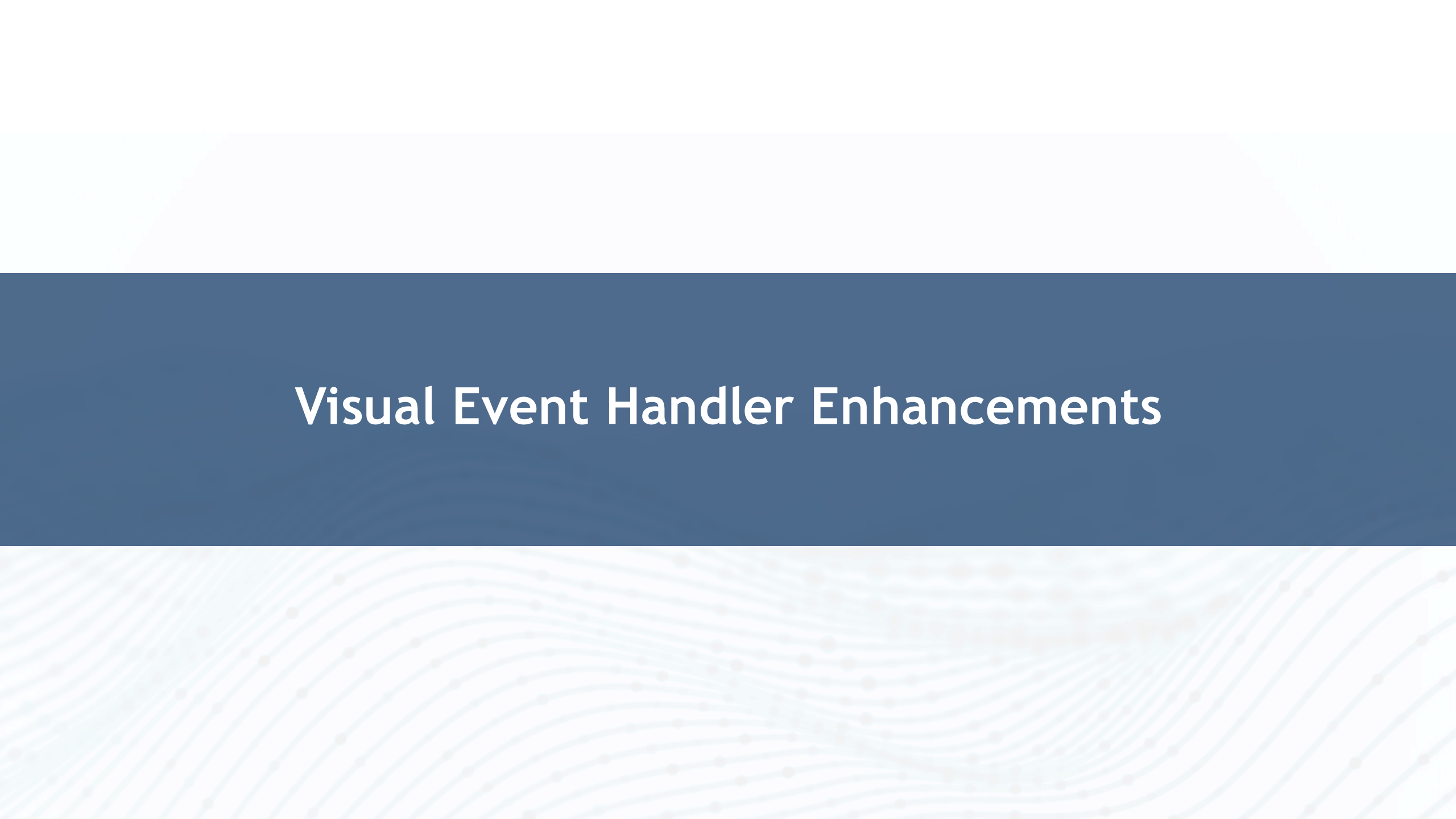
Start Over

LLM Playground

LLM Playground

- Support for Importing and Exporting conversation
 - Exported as Json files
- LLM Tool use now supported in any LLM
 - Bedrock
 - Azure OpenAI
 - If the LLM doesn't support Tool use, you will get an error

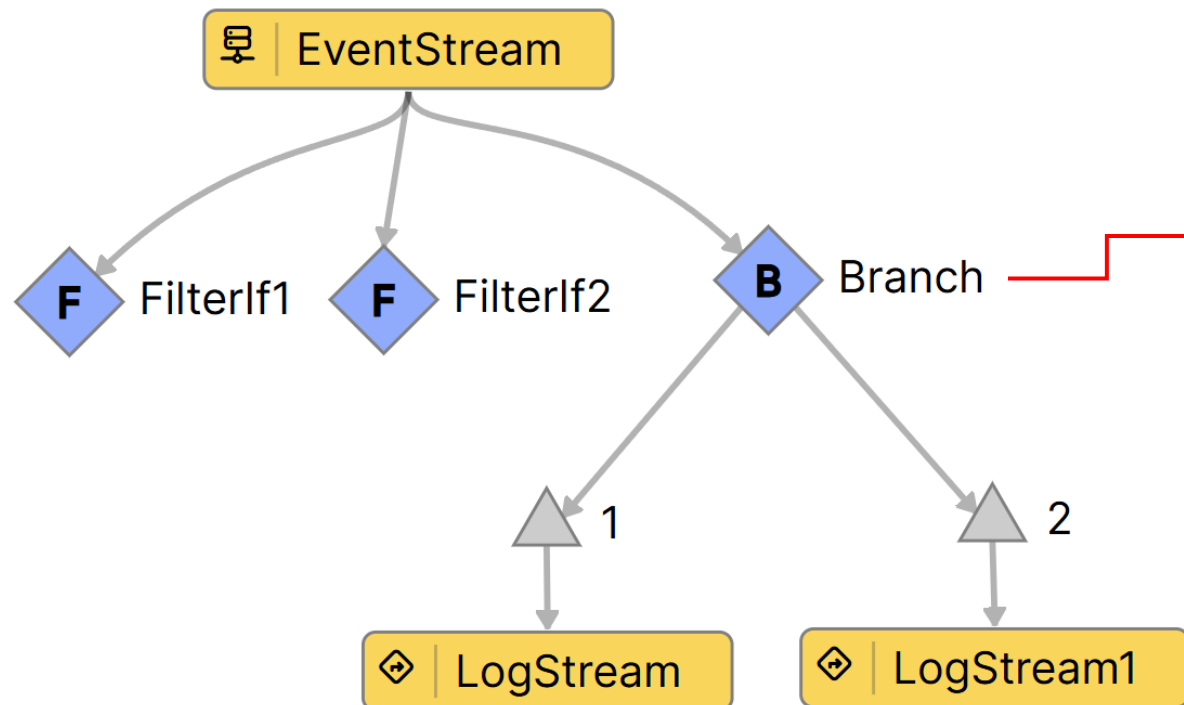




Visual Event Handler Enhancements

Activity Pattern Enhancements

- Branch Activity Pattern
 - Removes the need for multiple Filters



Edit parameter 'conditions' (an Array of Labeled Expressions)

A list of conditional expressions which will determine which outbound event fires.

[Add an Item](#)

Label	Expression
123	event.test = "123"
456	event.test = "456"

The branch has multiple conditions each will result in a separate output event named after the label

ClientBuilder Enhancements

Conversation Widget Enhancements

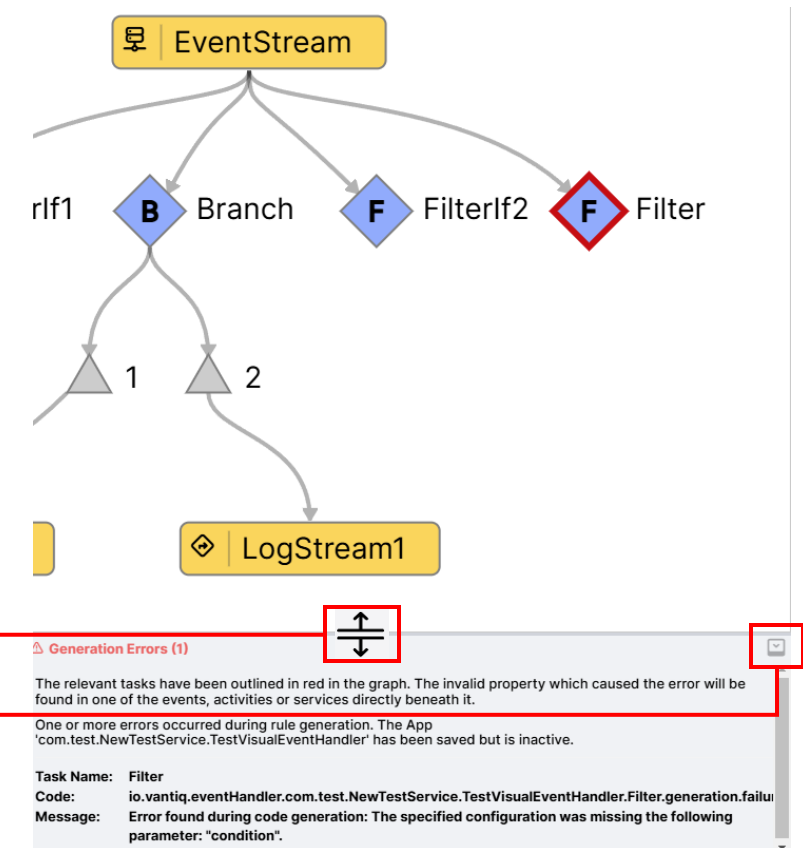
- Conversation Widget now supports Markdown
- Conversation Widget now supports standard Vail procedures as well as GenAI procedures

The slide features a dark blue horizontal band across the middle. Above this band is a light blue trapezoidal shape. Below the band is a background of light blue wavy lines with small orange dots. The text "IDE Enhancements" is centered in the dark blue band.

IDE Enhancements

Visual Event Handler and Gen AI Builder errors

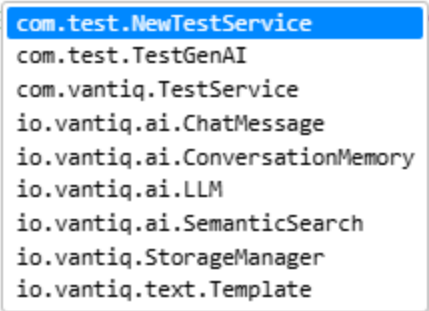
- The error pane in Visual Event Handlers and GenAI Builder is now collapsible and has a splitter



Miscellaneous

- VAIL Editor import service now includes built-in services

```
package com.test
import service
stateless PROCEDURE test_name_here()
```



Server Enhancements

Miscellaneous Server Enhancements

- Placeholder - Add built-in procedure to properly encode URL path
- Nodes now pay attention to the URI's path when targeting the local server. DO NOT include extraneous paths, e.g.
<https://dev.vantiq.com/ui/ide/index.html>
- System admins can set up resources to be automatically installed into every new org namespace. The resources are specified through a document (in system.documents) that is a zip containing an export of the resources - the file name must be `'org/newOrganizationResources.zip'`

Video Source

Video Source

- A VIDEO source represents a source of image data. This may be an online camera or a file. It connects to the Video Assistant
 - Stream data from the origination of the video. A request is issued to the video service periodically with the results producing an event
 - Query the video service for data. The video service is asked for content data and returns an image

Address or
filename

RSTP
Transport

Poll Interval –
0 no polling

Source: New Source

Test Data Receipt Publish to Catalog

General Properties

Keep Active ☒

Edit Config as JSON ☐

Message Type

Video Type -- Choose Type --

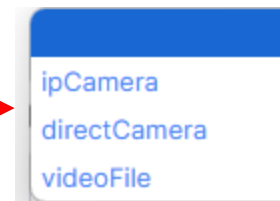
Video Address

RTSP Transport

Polling Interval

☐ http ☐ tcp ☐ udp ☐ udp_multicast

Video Type,
ipCamera etc.



Video Assistant

- Video Assistant is a service connector that ingests video streams and generates images.
 - The Video Source configures and polls the Video Assistant
- Video Assistant Service connection defined in a similar way to the GenAI Flow Service
 - Docker Compose or K8s install
 - Define Org or NS Service Connector

• Select `Administer/Advanced/Service Connectors` from the menu, click the `+ Connector` button, and specify the following values:

- **Name** – `VideoAssistantService`
- **Is Organization-wide** – checked (aka `true`).
- **Is Connector External** – checked (aka `true`).
- **Connector Hostname** – `localhost`
- **Connector Port** – `8890`
- **Request Timeout** – `120`

```
videoassistant:
  container_name: vantiq_video_assistant
  image: quay.io/vantiq/videoassistant:1.41
  restart: unless-stopped
  network_mode: "service:vantiq_edge"
```

Receiving Images

- Stream - define a polling interval and subscribe to events on the source via Visual Event Handler or Rule
- Query - use the SELECT syntax to pull images
 - `var someData = SELECT FROM SOURCE videoSource WITH contentType = 'image/png', resize = { maxHeight: 300, maxWidth: 400 } - Image resized and TempBlob`
- Streaming or Queries will return an object with three properties
 - fileType - the MIME type of the image entity
 - contentSize - the size (in bytes) of the image entity
 - contentRef - a ResourceReference to the TempBlob holding the entity contents

TempBlob's

- TempBlob's are in-memory representations of Binary Large Objects such as images
 - They have an expiration time of 2 minutes
 - Once they expire the ResourceReference will be invalid, and the image lost
 - TempBlob's can be saved as images or document
 - TempBlob's can be passed in REMOTE call if they are to be passed to an external system
- The amount of memory that is available to hold these TempBlob's is controlled by an Organization quota - `documentExpansion` – which is 0 by default

Saving Image to System.documents

- Use 'saveAs' to save image straight to system.document or system.images

```
var imageRef = "system.documents/myImage"  
var someData = SELECT ONE FROM SOURCE videoSource WITH saveAs  
= imageRef
```

- Image saved to documents etc.
- Use the 'processWith' to specify the name of a procedure that will be called to determine whether to save the received entity or to discard it
- Both of these are the same syntax as using SELECT against a remote source that returns an image

Saving TempBlobs in VAIL

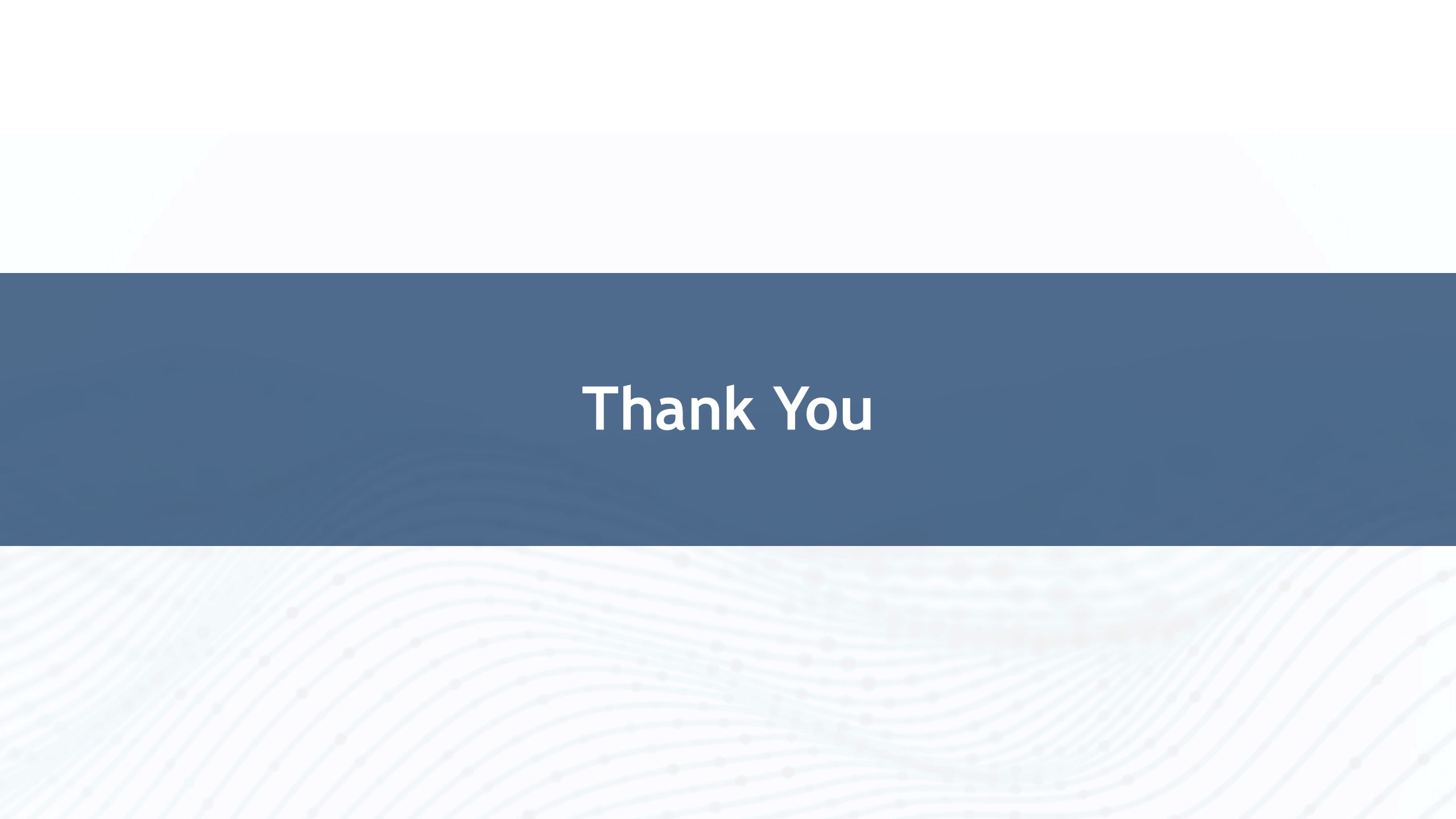
- 'SaveAs' and 'processWith' allow you to save the image to `system.document` via the `select` statement
- If you want to save it manually you can use a 'ResourceAPI.executeOp' statement

```
var imageData = select * from source videoSource
var tmpBlobRef = imageData[0].contentRef
ResourceAPI.executeOp{
    "op": "upsert",
    "resourceName": "system.documents",
    "object" : {name: "documentName", fromTemp: tmpBlobRef}
}
```

React Native Library

New React Native library

- Integrating VantIQ functionality into the React JavaScript framework is accomplished using the vantiq-react npm module
 - Authentication using either Keycloak or VantIQ Internal methods (only supported on Edge instances);
 - Push Notifications with built-in handling of location tracking;
 - Creation of new users using either Keycloak or Internal authentication;
 - Execution of streamed Procedures (i.e. executeStreamed);
 - Standard CRUD-oriented REST operations (e.g., select, insert, upsert, etc.);
 - VantIQ-oriented REST operations (i.e., publish, publishEvent, execute, executePublic)



Thank You