

VANTIQ 1.43 Update

VANTIQ Global Field & Partner Enablement

Agenda

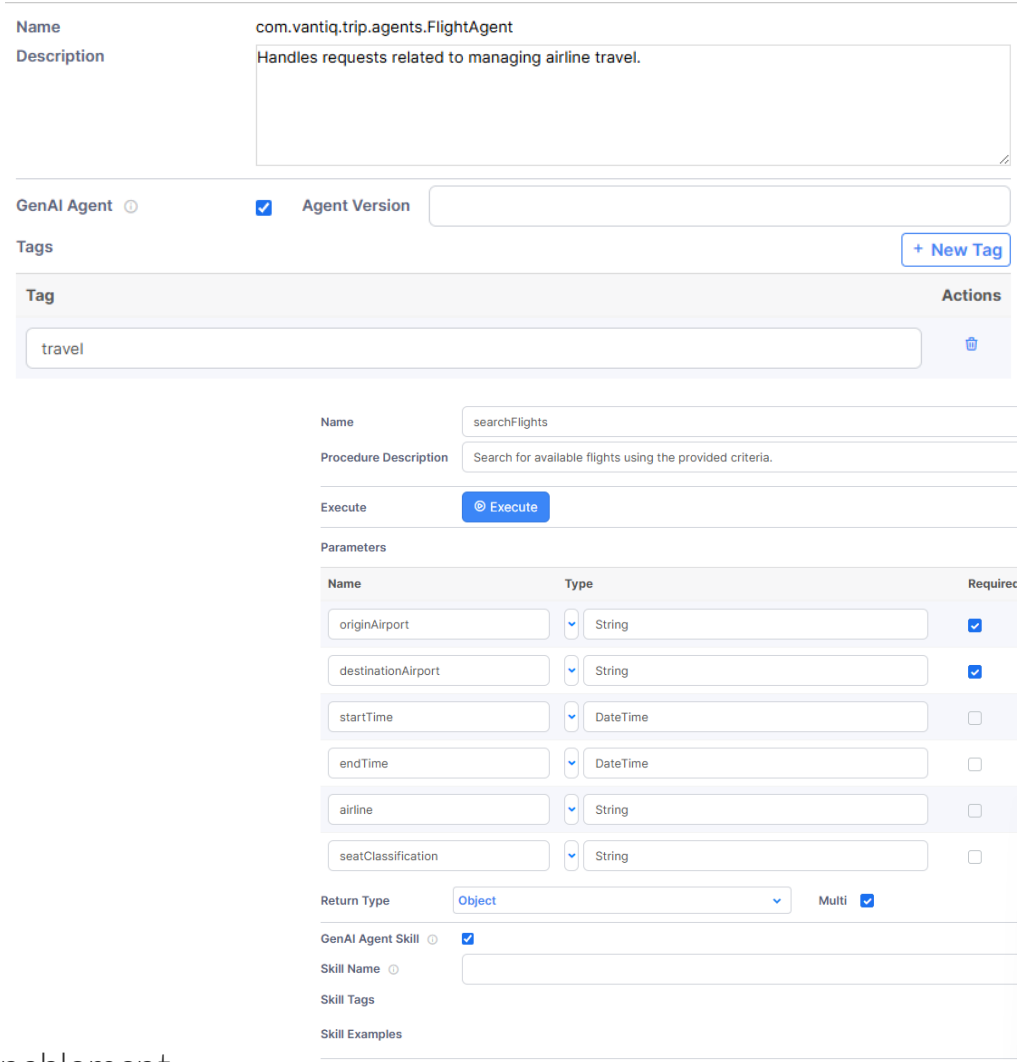
- Overview of New Features and Enhancements
- A2A and GenAI Agents
- GenAI Builder Enhancements
- GenAI Enhancements
- IDE Enhancements
- Open Inference Protocol Support
- Miscellaneous Enhancements

Overview of New Features and Enhancements

Usability Enhancements - Google A2A and Multi Agents (Preview Feature)

Increased developer productivity

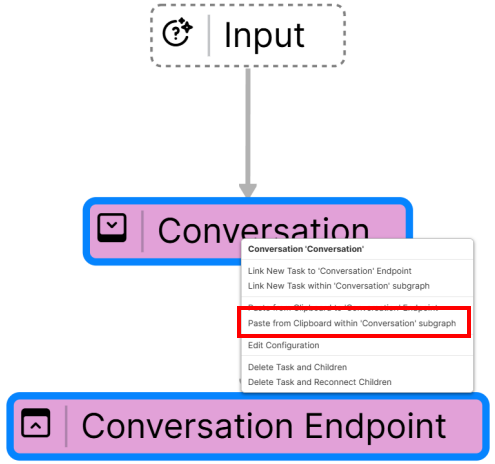
- Mark a Service as a GenAI Agent
  - GenAI Agents are now a first-class citizen
- GenAI Agent Model is based on the Google A2A specification
- GenAI Agents have Skills
  - Skills describe a specific capability or function that the agent can perform.
  - Skills are implemented as Procedures
- Framework to develop and execute multi agent systems



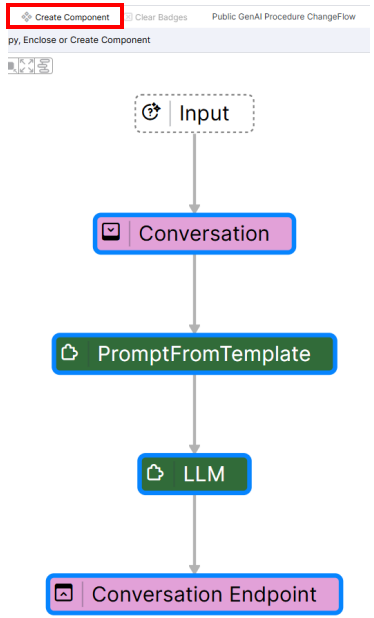
Usability Enhancements- GenAI Builder

Increased developer productivity

- Ability to enclose, or paste subgraphs into existing compound tasks



- Ability to create GenAI Components from subgraphs in the GenAI Builder

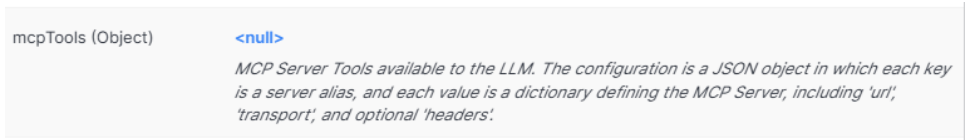


GenAI Enhancements – Several GenAI Enhancements

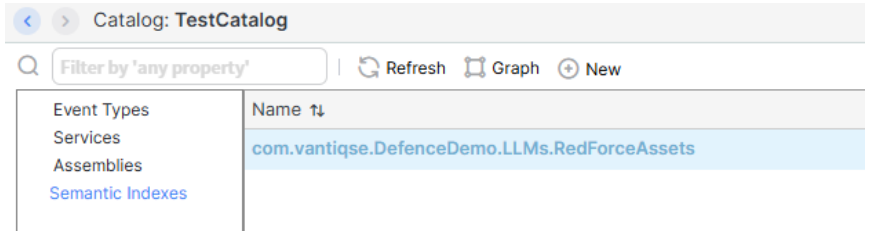
Increased integration capabilities



- Model Context Protocol (MCP) support
  - Can now configure the Tool activity pattern to integrate MCP-based tools



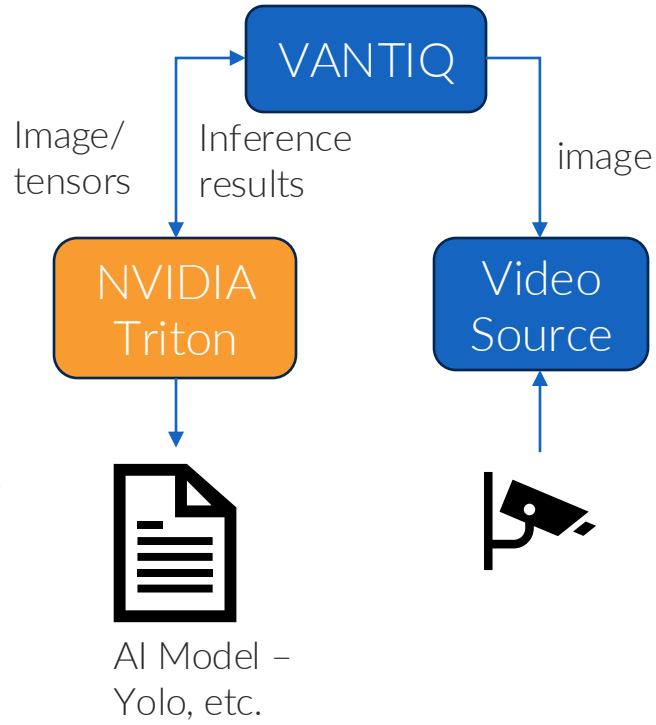
- Semantic Index Changes
  - Register in a Catalog or remotely accessible
  - Partition by user



Support for Open Inference Protocol

Increased integration capabilities

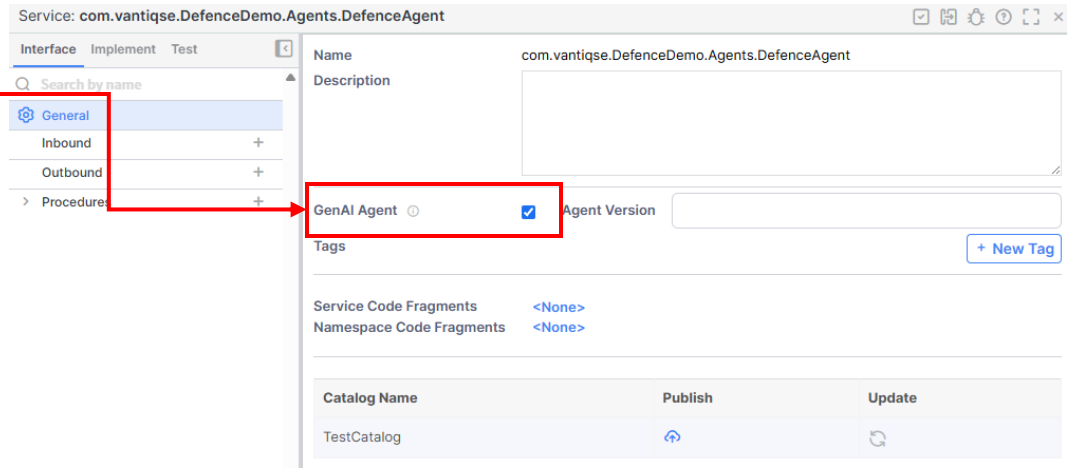
- The Open Inference Protocol (OIP) is a specification that standardizes communication between client applications and AI model inference servers.
  - Supported by several Inference servers – NVIDIA Triton, from example
  - Based on the Remote sources and utilizing the REST API
  - Pass tensors or an image to a model and return the inference results



A2A and GenAI Agents (Preview Feature)

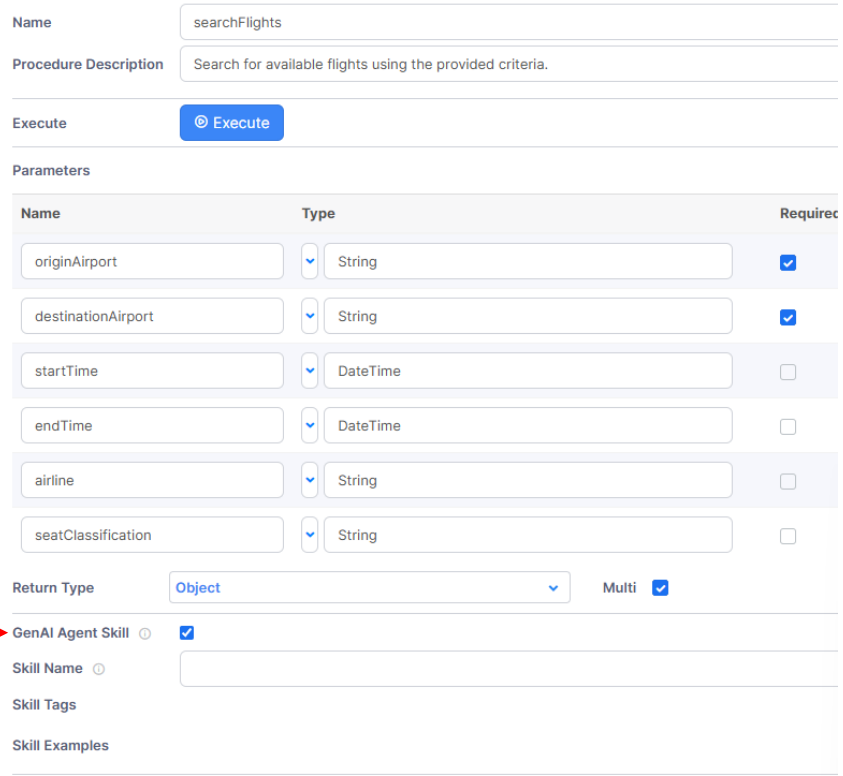
GenAI Agents and A2A

- GenAI Agents within Vantiq are based on the Google A2A specification - https://a2a-protocol.org/latest/specification/
- Before 1.43, it was recommended that a GenAI Agent be implemented as a Service. In 1.43, this is now been formalized.
  - Mark a Service as a GenAI Agent



GenAI Agents and Skills

- A GenAI Agent has Skills in the A2A specification
  - A Skill is a capability or function that an AI Agent advertises to other Agents or Clients in a multi-agent system
- Procedures can be marked as a Skill
  - On the interface definition of a Procedure check the GenAI Skill
  - Skills can have tags and examples
  - Procedure must be a public VAIL Procedure
    - Need to define a procedure comment
    - Need to define parameter descriptions
    - Same as Tools



Agent Card and Agent Discovery

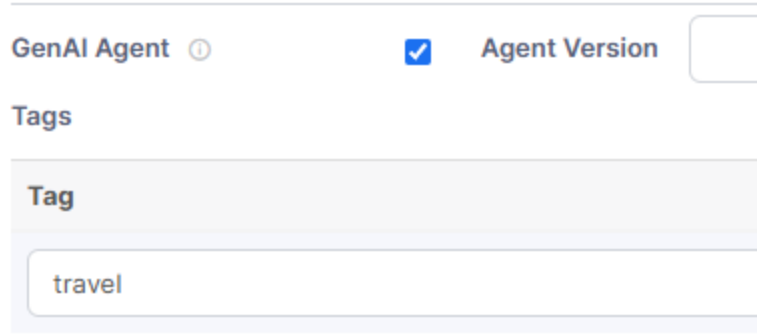
- A2A Agents must provide an Agent Card
  - The Agent Card is a document that describes the server's identity, capabilities, skills, etc.
  - Agent Cards can be generated using the '`io.vantiq.ai.A2A.getAgentCard(agentName String Required)`' built-in function
  - Or using the '`/api/v1/resources/services/<agentName>/_agentCard`' API

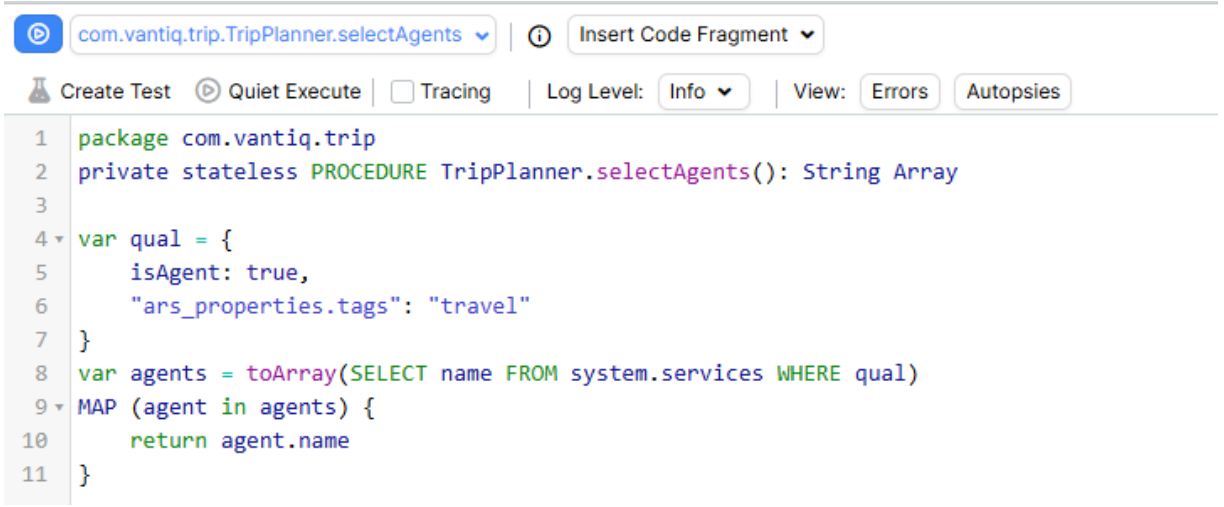
Execution of 'com.vantiq.trip.agents.FlightAgent.getMyCard' complete


```
{
  "name": "com.vantiq.trip.agents.FlightAgent",
  "description": "Handles requests related to managing airline travel.",
  "url": "https://test.vantiq.com/api/v1/resources/services/com.vantiq.trip.agents.FlightAgent",
  "version": "",
  "capabilities": {},
  "defaultInputModes": [
    "application/json"
  ],
  "defaultOutputModes": [
    "application/json"
  ],
  "skills": [
    {
      "id": "searchFlights",
      "name": "searchFlights",
      "description": "Search for available flights based on the given criteria",
      "tags": []
    }
  ]
}
```

Agent Card and Agent Discovery

- All A2A/Gen AI Agents are services and can be discovered by querying '`system.services`'
  - Use the GenAI Agents tags to group related Agents or form a team of Agents

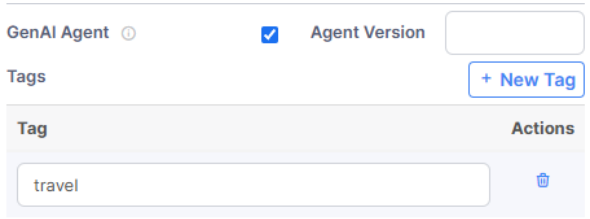


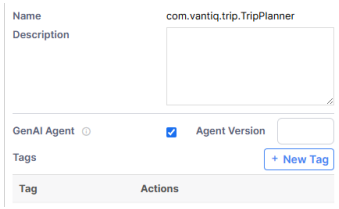


Agent Framework overview

- The Agent framework provides a way to invoke Agents using a common entry point and allows Agents to interact with users
  - The entry point is part of the framework and not visible
- The '`executePlan`' built-in function will call the Agent entry point
  - io.vantiq.ai.Agent.executePlan(planGraph io.vantiq.ai.PlanningGraph Required, userRequest Any, requestContext String): Any
  - This function executes a plan that is represented as a Directed Acyclic Graph (DAG)
- Agents will often need to interact with users to gather additional information. This is through the '`io.vantiq.ai.Agent.requestUserInput`' procedure
  - The developers will have to implement a temporary/bridge version in 1.43 – more on this later

Agent Framework overview





Client App

processRequest

a2a_requestUserInput
bridge procedure

Point Entry-Trip Planner Agent

processRequest calls io.vantiq.ai.Agent .executePlan() with a plan (DAG) containing a list of Agents

A2A Framework

Executes each step of the plan
- Async if possible
- Requests user input if required
- Calls the Agent using the A2A protocol

Flight Agent

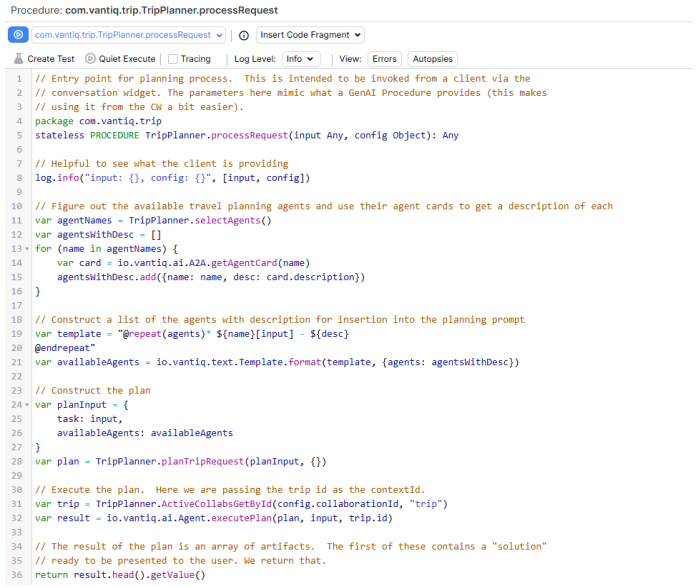
Hotel Agent

Car Rental Agent

Executes each step of the plan by
- Create skill plan – a2a_dispatchPlan
- Requests user input
  - Implicitly via the framework
  - Explicitly via a call to Agent.requestUserInput
- Execute a2a_dispatchSkill

Entry Point – TripPlanner

- The Entry Point service ‘TripPlanner’ is implemented by the developer
  - The process of creating the plan and calling the executePlan is via a user-defined procedure
  - This procedure is associated with a conversation widget



Entry Point – TripPlanner – User input

- The Entry Point service TripPlanner interacts with the user and requests additional input
  - The user must define the a2a_requestUserInput bridge procedure
  - This procedure is called automatically by the framework but needs to be implemented by the developer (will change later)
  - Uses the Callback or Notify mechanism to communicate with the Conversation Widget



Request User Input Flow

- The Flight, Hotel, and Car Rental Agents do not implement the a2s_requestUserInput
  - This is implemented internally
  - The requestUserInput is propagated up to the last Agent that implements the a2s_requestUserInput
    - This procedure can then interact with the user via the Callback mechanism
- Implementing the bridge a2s_requestUserInput is temporary for 1.43 and will not be needed in 1.44
- Only one request for input will be active at any point in time. If multiple agents need input, they will be processed one at a time.
  - This is handled by the framework automatically

Generated Procedures

- Marking a Service as a GenAI Agent will automatically generate two GenAI Procedures
  - a2a_dispatchPlan - generates a plan for which skills to execute on an Agent
  - a2a_dispatchSkill - calls a particular skill
- These are internal procedures and should not be modified
- These procedures are called by the A2A Framework and should not be called directly
- They are exposed, as there may be rare occasions that require changes to these procedures
  - They can/will be regenerated by checking/unchecking the ‘GenAI Agent’ checkbox

Executing Plans

- There are various planning algorithms that can be used to define a ‘plan’
  - Plan and Execute
  - Reasoning without Observations (ReWOO)
  - LLMCompiler
- It is up to the developer as to which approach to take to generate the initial ‘plan’ that is then executed by the ‘`io.vantiq.ai.Agent.executePlan`’ built-in
  - The structure of the plan is documented and there are several schema types associated with the DAG - `io.vantiq.ai.PlanningGraph`, `PlanningGraphNode` and `PlanningGraphEdge`
- Examples and GenAI component will be provided to make this easier – in 1.43 examples only

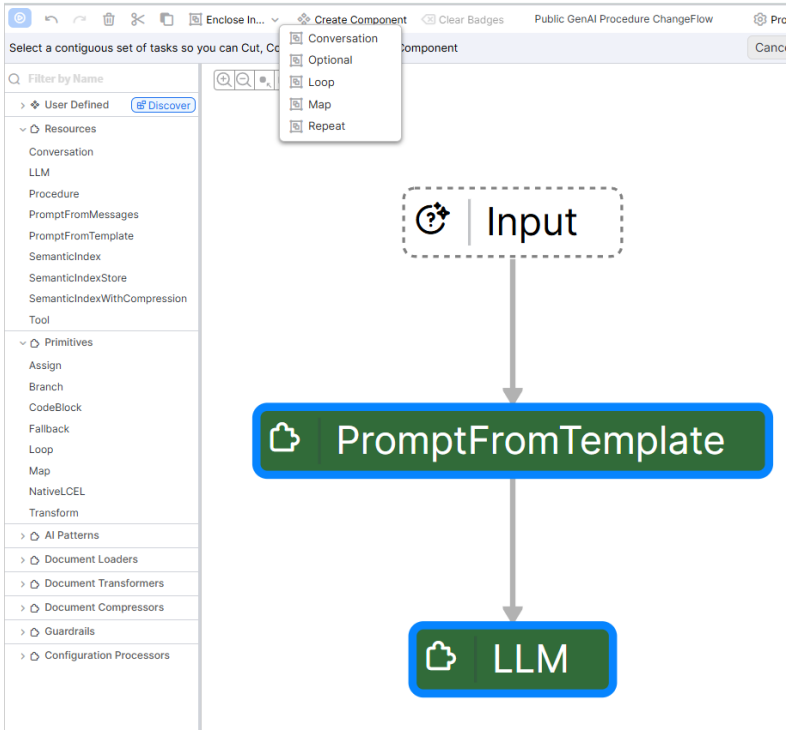
Other A2A resource

- There are a few additional A2A Framework types and built-ins in 1.43
  - Types
    - io.vantiq.a2a.Task
    - io.vantiq.a2a.Artifact
  - Built-in
    - Agent.getCurrentTask()
      - This is used when an agent that is being called by the framework wants access to the current task "context" (useful for passing entity ids).
    - Agent.requestUserInput(requestData Any Required, requestContext Any): Any
      - This is used when an agent that is being called by the framework wants to request user input explicitly
- These are documented, but their usage will probably be limited in the 1.43

GenAI Builder Enhancements

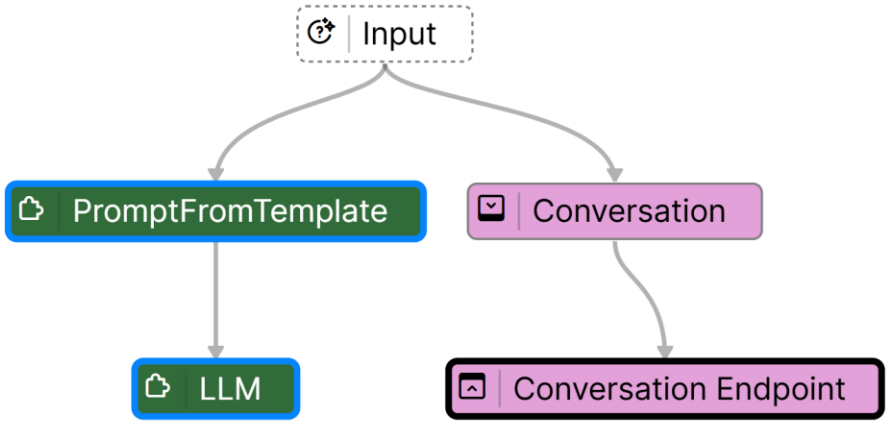
Enclosing Task in a Conversation etc

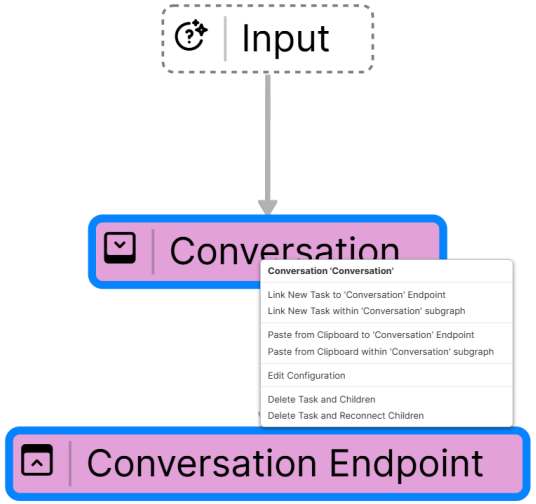
- Before 1.43, if you wanted to enclose a set of tasks in a ‘Conversation’, ‘Loop’, etc, you had to delete them and start again
- Now you can enclose multiple tasks
  - Shift Select multiple tasks
  - Click on the ‘Enclose In…’ menu option

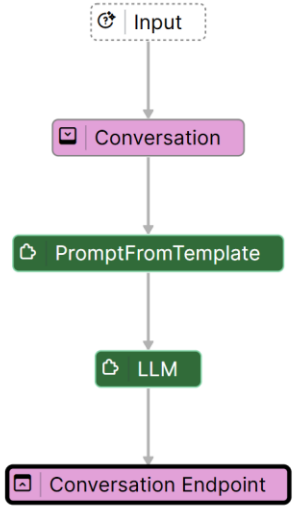


Cut and Paste within a subgraph

- You can now cut and paste a set of tasks into a ‘Conversation’, ‘Loop’ etc
  - Select multiple tasks and ‘Cut’
  - Select the ‘Conversation’ or ‘Loop’ etc and right click
  - Select ‘Paste from Clipboard within ‘…’ subgraph

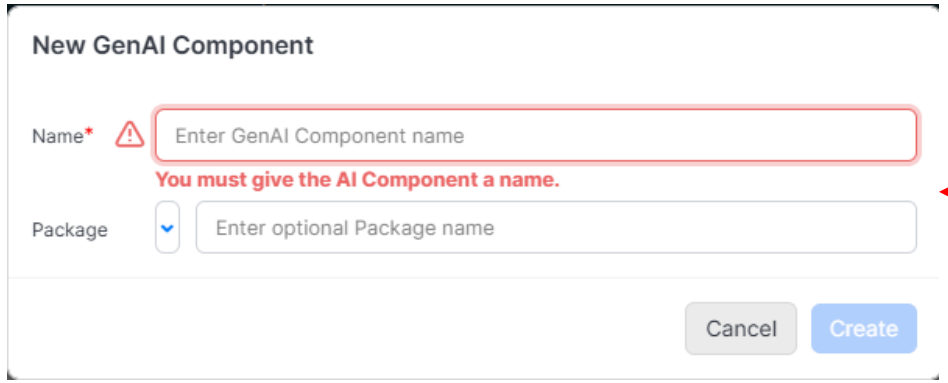




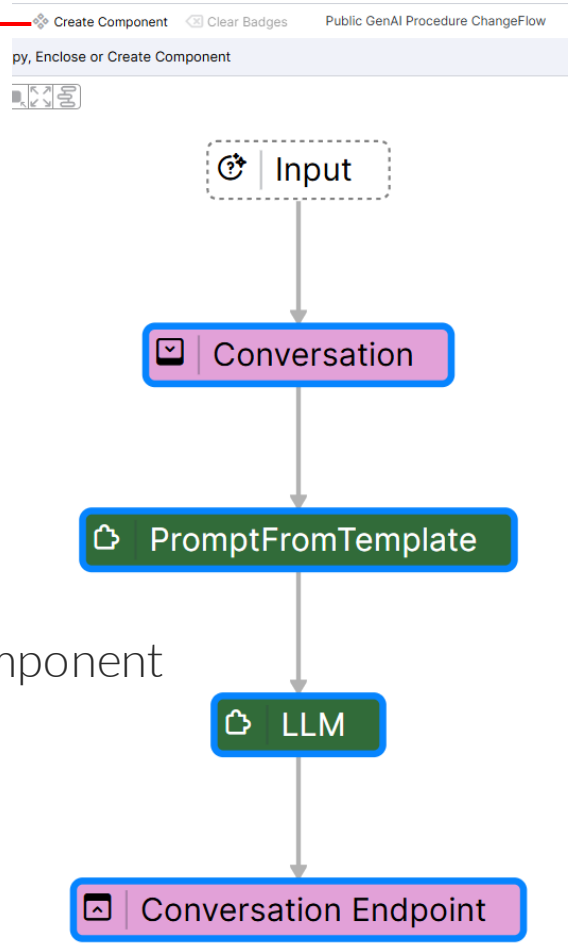


Create GenAI Components from within GenAI Builder

- Before 1.43, you could only create GenAI Components by creating a new GenAI Component
- Now you can select a sub graph of tasks and create a GenAI Component from these tasks



- Select subgraph
- Click Create Component



Other GenAI Enhancements

MCP Support

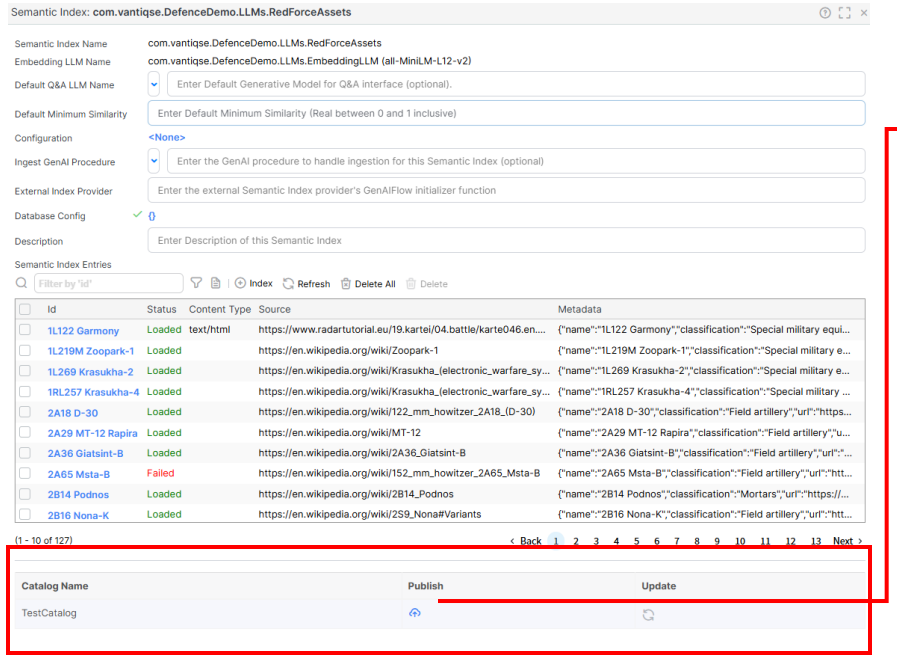
- GenAI Flows can now integrate with Model Context Protocol (MCP) based tools
- The Tool task has been enhanced to support MCP tool calling

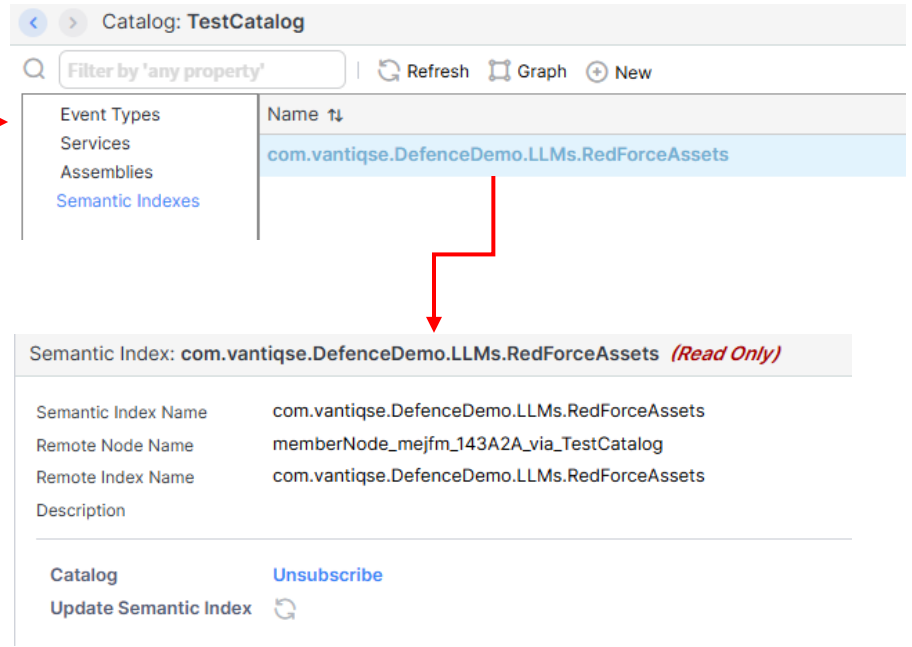
mcpTools (Object)	<null> *MCP Server Tools available to the LLM. The configuration is a JSON object in which each key is a server alias, and each value is a dictionary defining the MCP Server, including 'url', 'transport', and optional 'headers'.*

```
{
  "mcpTools": {
    "mcpServer1": {
      "url": "https://mcp-server1.example.com/mcp",
      "transport": "streamable_http",
      "headers": {
        "Authorization": "Bearer @secrets(MCPKey_Server1)"
      }
    },
    "mcpServer2": {
      "url": "https://mcp-server2.example.com/mcp",
      "transport": "streamable_http",
      "headers": {
        "Authorization": "Bearer @secrets(MCPKey_Server2)"
      }
    }
  }
}
```

Semantic Index Enhancements

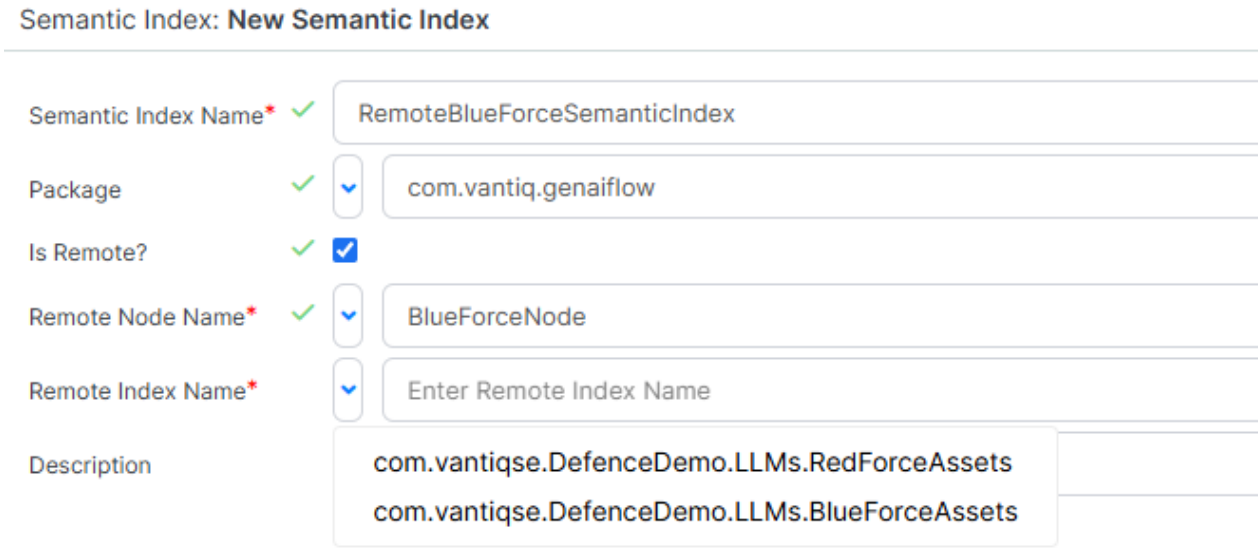
- Semantic Indexes can now be registered in a Catalog
  - Once a namespace is connected to a catalog the ‘Publish’ option will be available in the Semantic Index pane





Remote Semantic Indexes

- Can configure a Semantic Index to be Remote



- Similar to the Catalog approach but without being registered in the catalog – similar to a ‘PROCESS BY’ clause when executing a procedure

Catalog and Remote Semantic Indexes

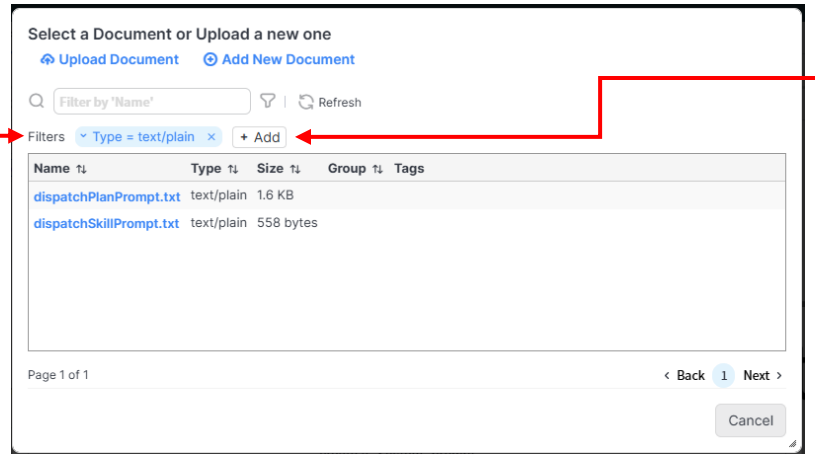
- Semantic Indexes that are either registered in a Catalog or accessed via the Remote options are effectively read-only to the client namespace
  - No support for add or view entries in the index
  - Only supports queries against the index
- If you want to allow a remote namespace to update the Semantic Index, then this should be done through a service in the namespace that hosts the Semantic Index and registering this service in the Catalog or accessing it through a PROCESS BY clause.
  - The service would implement the updates to the index

Semantic Index Partitioning

- When creating a Semantic Index, you can choose to ‘Partition By User’
  - Entries in the semantic index will be partitioned by the user id of the user who loaded the documents
  - When querying the index, they will only see documents that they loaded

Changes to Prompt activity patterns

- PromptFromTemplate Changes
  - Template Document - Only list text/plain by default and also adds filters



Ability to adjust the filter

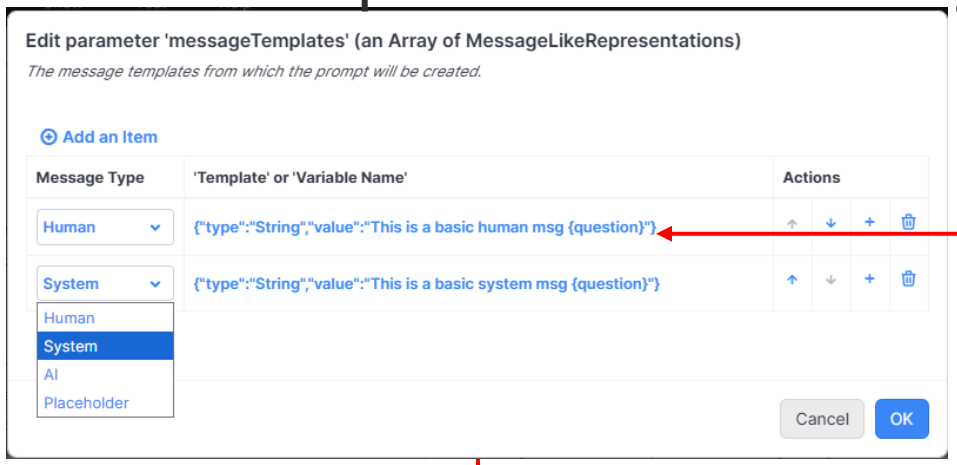
  - New option – ‘Template URL’ load template from a URL

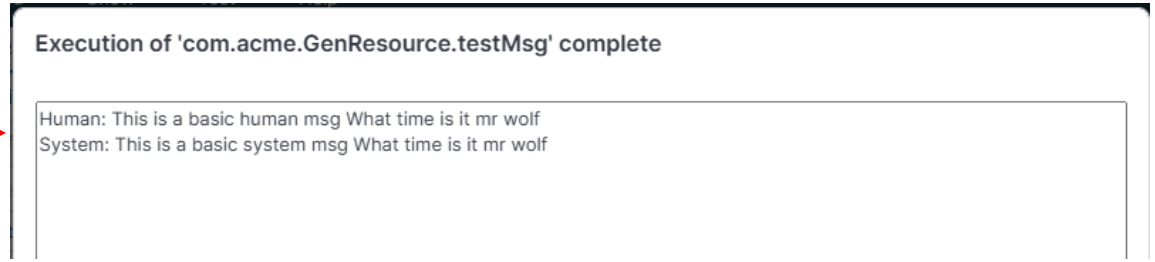
Changes to Prompt activity patterns cont.

- New PromptFromMessages
  - Generates a PromptValue from an array of messageTemplates

Message Types

Template String/Doc/URL





Tool Parameter Override

- There is a new runtime parameter – toolParameterOverride that is a json object with the following properties.
  - `tool` -- The name of the tool that the parameter belongs to.
  - `name` -- The name of the parameter to override.
  - `value` -- A substring to match against the parameter value selected by the LLM. If a match is found, the override is applied.
  - `override` -- The value to use instead of the value returned by the LLM.
- This can be used to override Tool parameters at runtime, for example, to provide a value for an optional parameter the LLM did not set, to replace a value it selected or to hide sensitive information (PPI) from the LLM

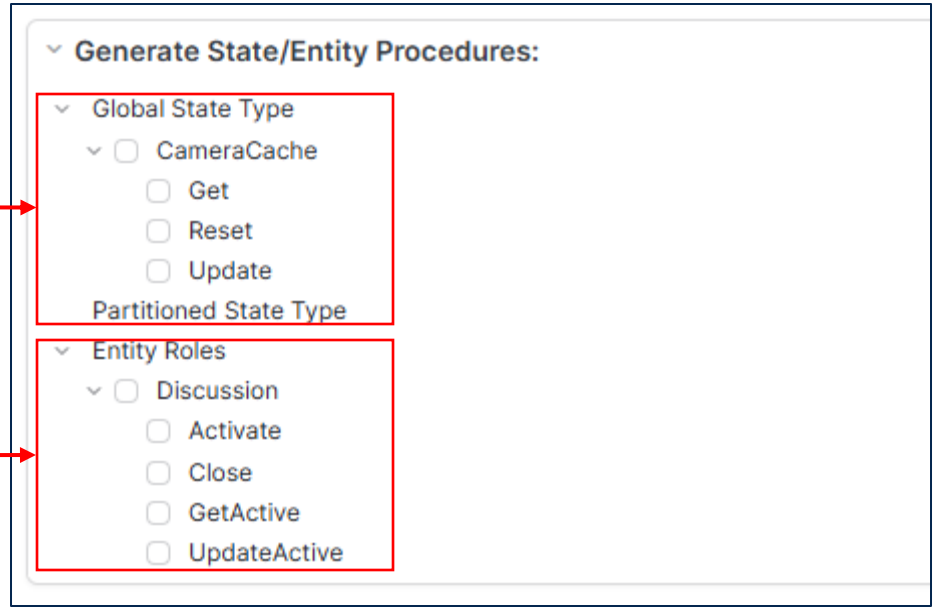
IDE Usability Enhancements

Generating State & Entity Role Procedures

- There is a new way to generate state and Entity Role accessor procedures
  - If you have State or Entity Roles defined on a service, you will see the new tree to generate the accessor procedures

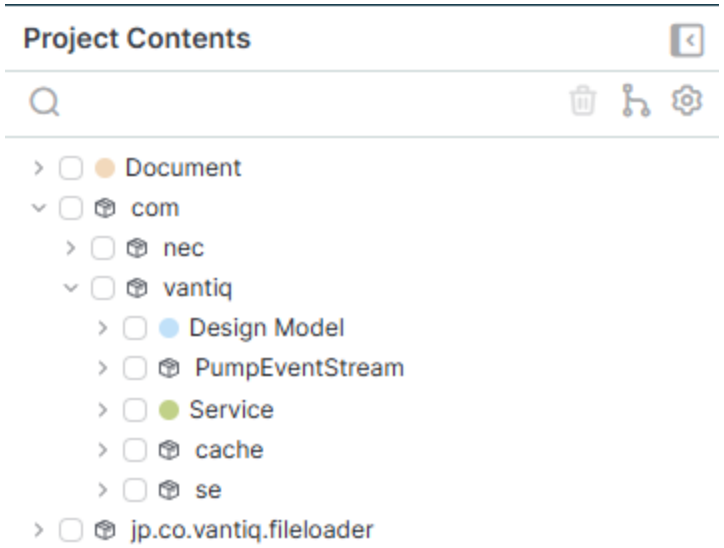
State-related procedures

Entity Role-related procedures

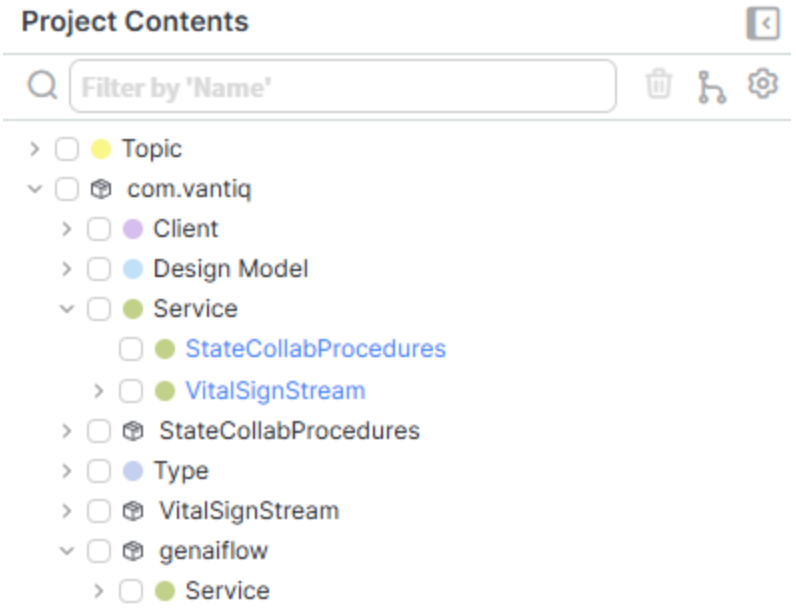


Streamline the Project Content Search box

1.42 – Click on the search icon to activate/deactivate the search. It wasn't obvious and not the way other search's worked



1.43 – The Filter box is always visible and doesn't obscure the other options. The x will appear to clear the filter when text in the filter input



Service Builder Changes

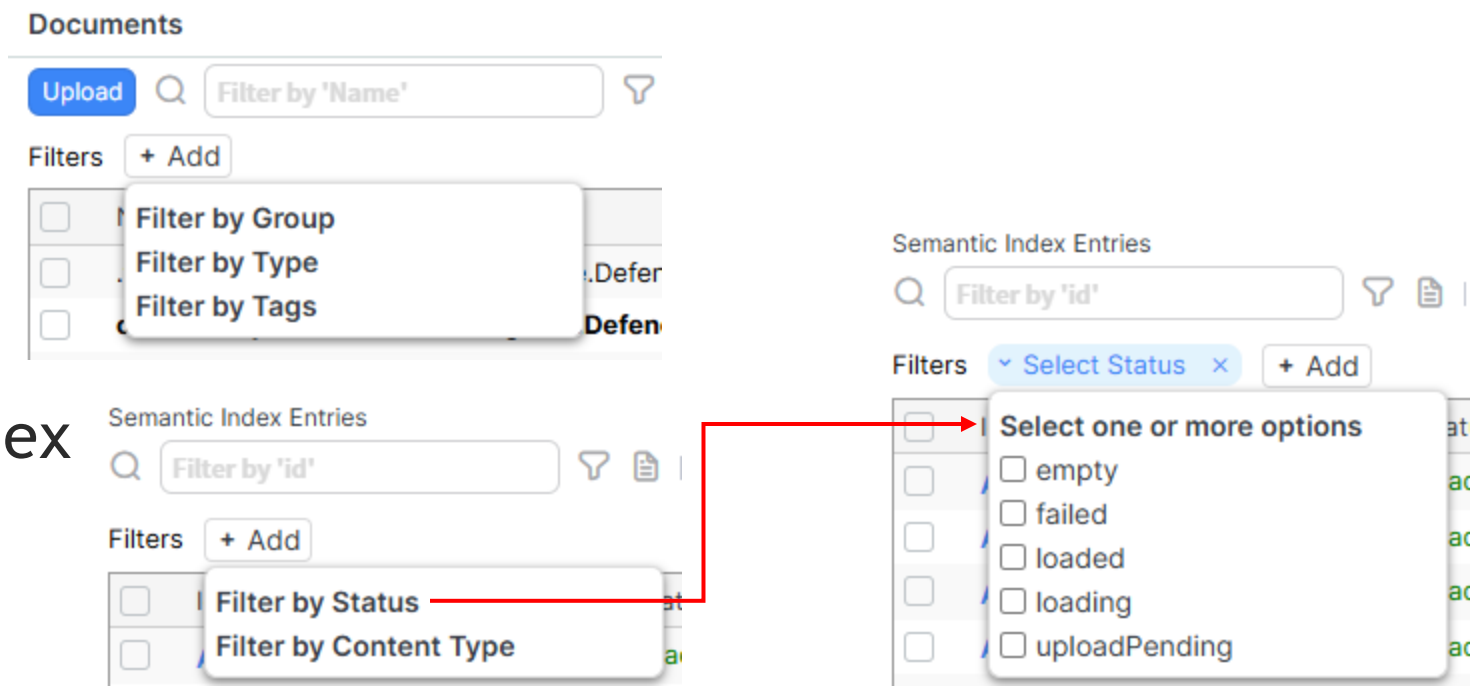
- Deleting procedures will now require a Save to be more consistent with other operations
  - Discard will put back deleted procedures
- Embedded GenAI Flows can now be executed – GenAI Procedures generated from the Visual Event Handler Activity Pattern GenAIFlow
  - This is to help with debugging

Visual Event Handler - Saving

- Layout changes, such as zooming and moving nodes, no longer dirties the Visual Event Handler – Save is not enabled
- Layout changes are only saved when material changes to the Visual Event Handler or tasks have taken place
  - Change Visual Event Handler properties
  - Change Tasks properties

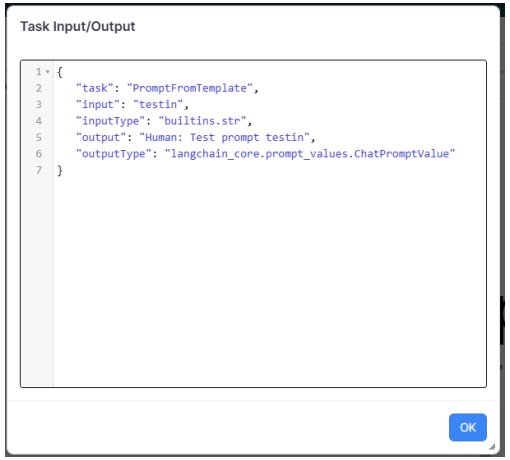
New Filter Options

- In several places in the IDE, the filtering has been enhanced by clicking on the funnel
  - Documents
  - Semantic Index



Other IDE Changes

- New shortcut keys for Flexgrid control
  - Win : Ctrl-K/Mac : Cmd-K - Close the current pane/tab
  - Win : Ctrl-Shift-K/Mac : Cmd-Shift-K - Close all open panes/tabs
- New IDE option – ‘Show Generated Resource’ this will disable/enable showing of generated resources in the project contents
- Task Output can now be expanded and copied



Open Inference Protocol Support

Open Inference Protocol (OIP)

- The Open Inference Protocol (OIP) is a specification that standardizes communication between client applications and AI model inference servers
  - OIP defines two wire protocols – REST & gRPC
- OIP supported by several Inference Servers
  - KServe
  - NVIDIA Triton
  - Seldon
  - Seldon Core
  - OpenVino
  - AMD
  - TorchServe
- See https://github.com/kserve/open-inference-protocol
- Documentation and extensive tutorial provided

OIP Integration

- Integration via the Rest Protocol
  - Use a Remote Source
  - Specific the URI for the Inference Server
  - Set the ‘openInference’ property to true to enable binary tensors
    - Must check if the OIP server supports binary tensors
    - Passing tensor data in binary format is more efficient and transparent if the server supports it

```
{
    <omitted properties>,
    "requestDefaults": {
        "openInference": true
    },
    "uri": "http://localhost:8080"
}
```

Example VAIL code

```
// Assuming that IpCamera is a VIDEO source
var imageFeed = select one from source IpCamera with resize = { maxHeight: 640, maxWidth: 640 }

var payload = {
  inputs: [
    {
      name: "image",
      shape: [ 1, 1 ],
      datatype: "BYTES",
      data: [ Utils.packageReference(imageFeed.contentRef)]
    }
  ]
}

// Assuming that the TritonServer is configured with openInference = true
var modelResponse = select one from source TritonServer with path = "/v2/models/myYoloModel/infer",
                                        method = "POST", body = payload
```

Flatten and Reshape

- The Remote source will automatically flatten and reshape the tensors when sending and receiving
  - If you do not wish that to happen, you can set the openInferenceReshape = false
  - Two built-in procedures are provided to allow you to perform the flattening and reshaping manually
    - io.vantiq.ai.OpenInference.reshape(data, shape)
    - io.vantiq.ai.OpenInference.flatten(data)

```
// Assuming that the TritonServer is configured with openInference = true
select one from source TritonServer with path = "/v2/models/mymodel/infer",
                                method = "POST", body = inferencePayload, openInferenceReshape = false

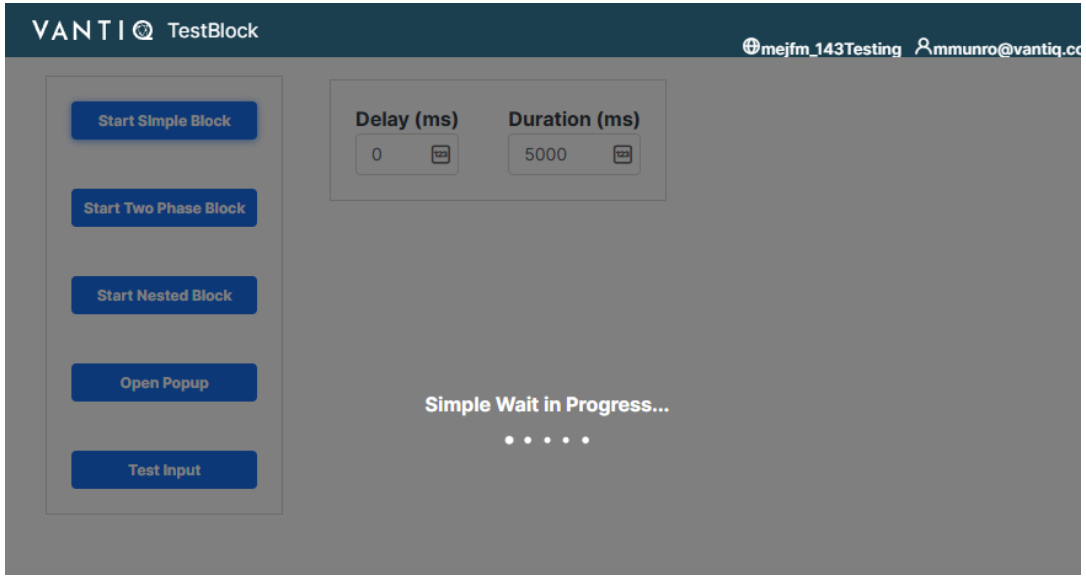
// Response tensor is not reshaped
{
  "model_name": "mymodel",
  "model_version": "1",
  "outputs" : [
    {
      "name" : "result",
      "shape" : [ 2, 3 ],
      "datatype" : "INT32",
      "data" : [ 1, 2, 3, 4, 5, 6 ]
    }
  ]
}
```

Miscellaneous Enhancements

Spinner support in the ClientBuilder framework

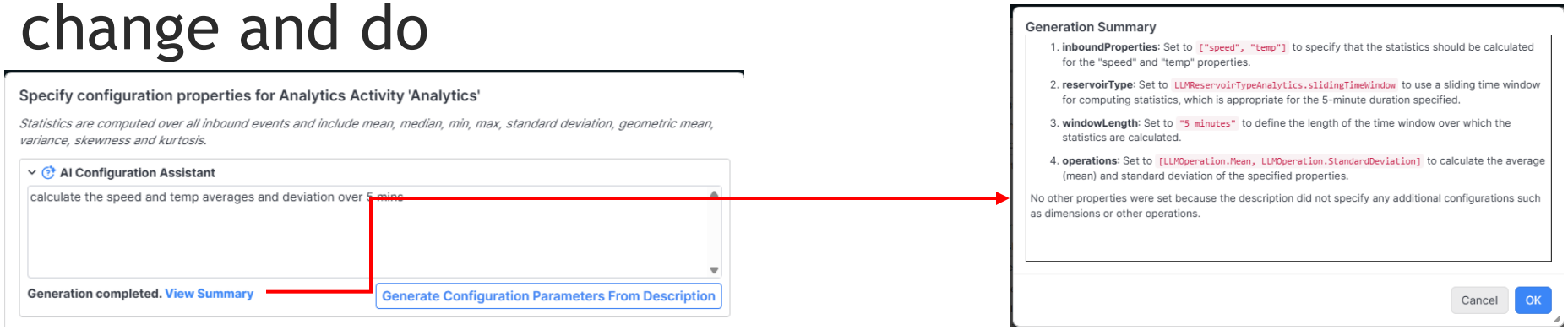
- New Client functions to block input and complete

```
5   function Client_Start_Button4_onClick(client, page, extra) {
6
7       client.blockInput("Simple", "Simple Wait in Progress...");
8
9       client.setTimeout(function () {
10          client.blockInputCompleted("Simple");
11      }, client.data.duration);
12  }
```



Visual Event Handler - task AI Configuration Assistant changes

- You can now see a summary of what the AI Assistant decided to change and do



- Especially useful when the assistant doesn't do anything

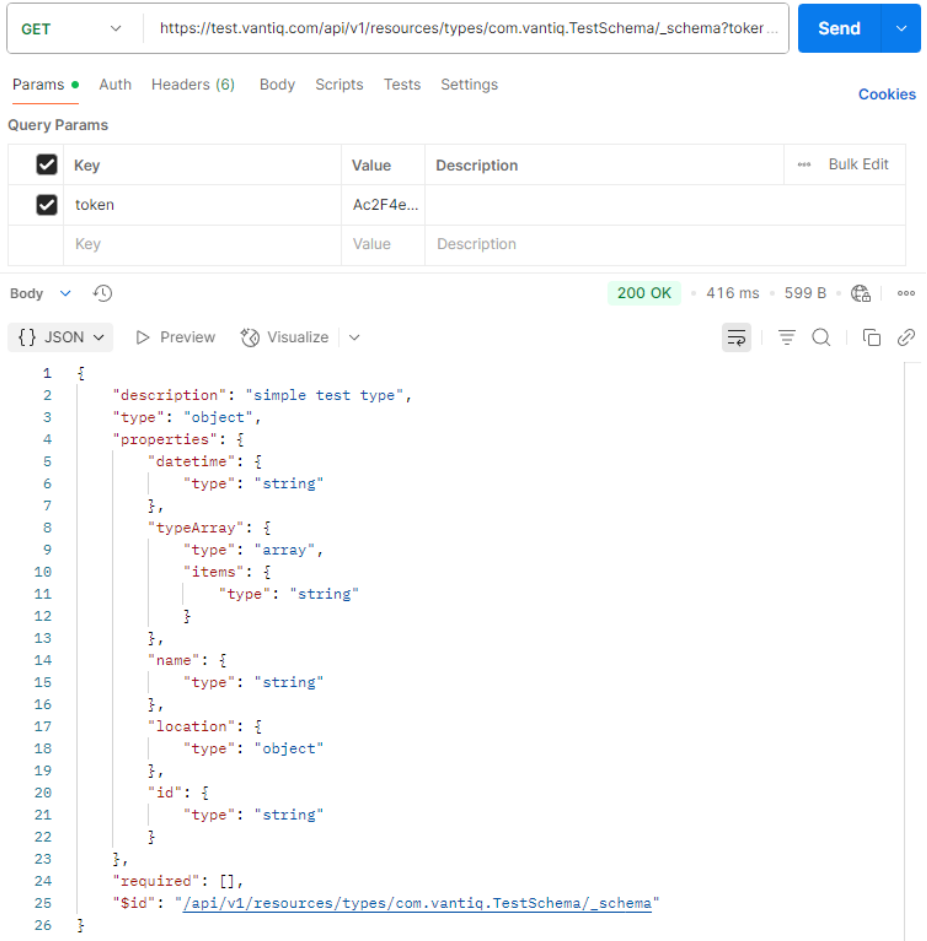


Server Enhancements

- ‘io.vantiq.text.template’ now supports ‘@repeat(items)’ & ‘@endrepeat’
  - Repeats a block of text based on an array passed as ‘items’
  - Special ‘${value}’ substitution for each element of the array in ‘items’

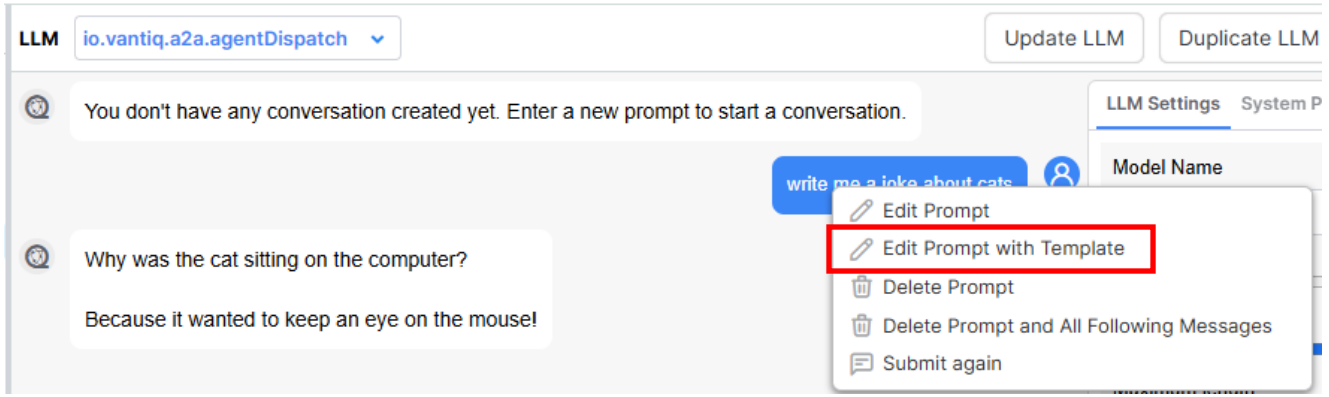
Server Enhancements cont.

- Types now have an option to generate a JSON Schema for a type using the API.
- Either the ‘selectOne’ operation or
- the URL ‘resources/types/<typeName>/_schema’
  - HTTP GET - `https://test.vantiq.com/api/v1/resources/types/com.vantiq.TestSchema/_schema`

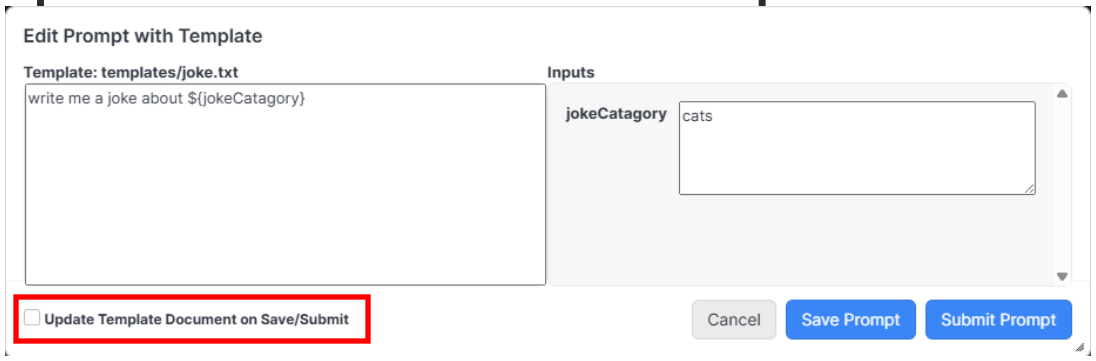


LLM Playground

- If a prompt was generated from a ‘Template’ it can be edited in the template editor



- Can save document templates within the template editor



Deprecated/Remove Features

GenAI Builder - Guardrails

- GuardrailsAIServer GenAI Component has been removed
  - This was due to issues with the underlying library
  - The GuardrailsAI is still available with the predefined validations
  - NeMo is still available for custom validation behavior

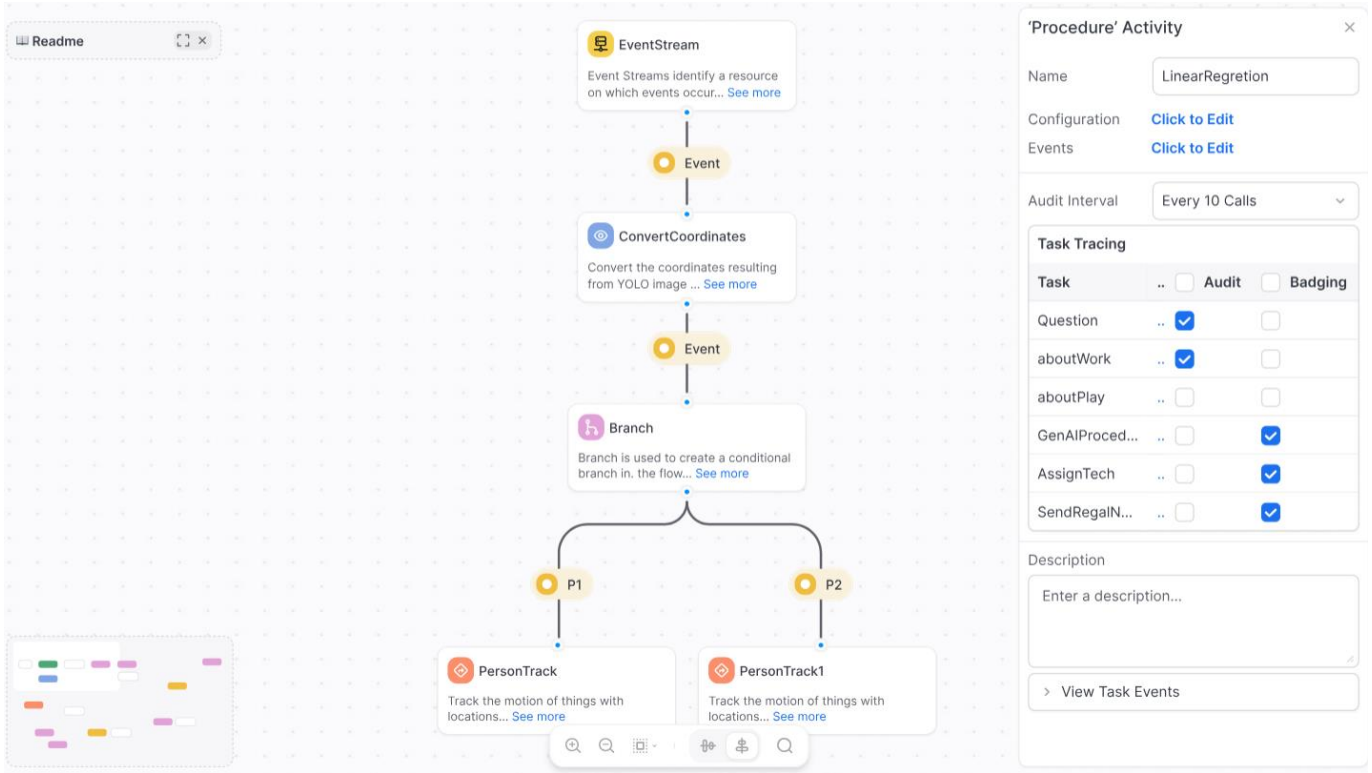
1.44 Main Features

VIA and A2A - P1

- Vantiq Intelligent Assistant VIA - work was started in 1.43 but not completed.
  - First release will be Client and Service Builder focused
- VIA will also be driving a lot of A2A and Multi Agent feature in 1.44
  - Expose more of the A2A framework
  - Client Builder integration via a new Widget – ‘Discussion Widget’

Visual Event Handler and GenAI Builder UI refresh – P2

- Update the visual appearance of the Visual Event Handlers and GenAI Builder
  - More modern look and feel
  - Minor usability changes



Other enhancements

- Most other enhancements will be in support of the P1 and P2 features
  - Tree View Widget
  - GenAI Planning Components
- TensorFlow and Vision capabilities will be removed – These have not been available for years and will now be removed from the document etc. The Video source and OIP replace these capabilities.

Thank You