

HALOITSM

Everything You Need to Know About AI in HaloITSM



HALOITSM

- Halo’s AI Philosophy3**
 - Gartner Recognition3**
- AI Security4**
 - Halo’s OpenAI Connection4**
 - Customer’s OpenAI Connection5**
 - Customer’s Azure OpenAI Connection6**
 - AzureAI Search7**
- AI Data Architecture9**
- AI Functionalities and Configuration 10**
 - AI connection methods 10**
 - Embeddings 12**
 - AI Insights 14**
 - AI Suggestions 16**
 - Article Creation 19**
 - Thank You Detection 21**
 - Actions 22**
- Pricing for AI in HaloITSM 32**

Halo's AI Philosophy

At Halo we believe AI will revolutionise Service Management, and the world, offering organisations an unmatched ability to increase efficiency and streamline every part of the customer experience. AI is at the forefront of every customer conversation we have; every CIO wants to adopt AI- however, there is widespread concern about the ROI from AI because almost every SaaS company has placed it behind a 'paywall'. At Halo we are completely committed to our philosophy of all-inclusive pricing, AI will always be included as part of the standard Halo license cost, even as we continue to innovate and set the standard for how AI can be used to optimize Service Management.



Paul Hamilton, CEO/Founder of Halo

Gartner Recognition

In 2024, Halo Service Solutions was recognised by Gartner® in the Gartner Magic Quadrant for Artificial Intelligence in IT Service Management.

AI Security

Halo's OpenAI Connection

Data Transmission and Data Processing:

- **API Connection:** The ITSM system connects to Halo's OpenAI integration via an API. Data is sent over this connection for processing.
- **Encryption:** Typically, data is transmitted over secure connections using encryption protocols like TLS to ensure data privacy and security during transmission.
- **Vectorization:** The data sent to OpenAI is used for vectorization, a process that converts text into numerical vectors for machine learning purposes.
- **Temporary Storage:** When using all AI functionalities in Halo, except for the virtual agent, data is processed without being stored in memory or in a transient storage location. For interactions involving the virtual agent, data is retained until the chat session is completed, after which Halo will remove the temporarily stored data. As with the rest of OpenAI's platform, data and files passed to the OpenAI API are never used to train OpenAI's models.

Data Storage:

- **Temporary vs. Permanent Storage:** Depending on the service's architecture and policies, data might be stored temporarily during processing. Permanent storage of data typically does not occur unless specified as part of the service agreement.
- **Data Retention Policies:** The processing of data required between Halo or the underlying OpenAI service is done through an endpoint configured for zero data retention between Halo and OpenAI, governing how long data is stored, if at all. This zero data retention policy applies to all AI functionalities in Halo, with the exception of the virtual agent feature where Halo will remove temporary stored data from OpenAI as soon as the interaction is completed.

Access to Data:

- **Access Restrictions:** Access to data is generally restricted to authorized personnel only, which might include system administrators or specific employees within Halo with a need-to-know basis.

- **Customer Control:** Customers should have control over their data, including the ability to delete or request data deletions if stored beyond processing requirements.
- **Data Privacy Compliance:** Halo should comply with relevant data protection regulations (e.g., GDPR, CCPA) to ensure user privacy and data security.

Security and Compliance:

- Enabling this connection enables Halo's OpenAI to become a sub processor of the data. All data remains in the Halo's database which follows Halo's Security policy, GDPR policies etc.

Customer's OpenAI Connection

Data Transmission and Processing:

- When data is sent to OpenAI's API, it is transmitted over a secure connection (typically HTTPS) for processing.
- The data is used to generate a response, such as vectorization or language model output, based on the request made.

Data Storage:

- **Temporary Storage:** OpenAI processes the data temporarily in memory to generate the response. This is necessary for the API to function and return results. This applies to all AI functionalities with the exception of virtual agent.
- **Permanent Storage:** This will only apply when deploying the virtual agent functionality in Halo. OpenAI does not retain or store user data or API requests permanently. Once the processing is complete and the response is generated, the transaction data is removed as Halo will remove temporary stored data from OpenAI as soon as the interaction is completed.

Data Access:

- **Access by OpenAI:** The data is not accessible to OpenAI staff after processing unless there is a specific agreement or requirement for logging, auditing, or troubleshooting purposes. OpenAI prioritizes user privacy and data security.

- **Access by Third Parties:** The data is not shared with third parties unless explicitly agreed upon by the user or required by law.

Privacy and Security:

- **Privacy Policies:** OpenAI's privacy policy and terms of service govern the handling of data, ensuring compliance with privacy regulations and industry standards.
- **Data Anonymization:** OpenAI often implements data anonymization techniques to prevent any association of input data with individual users or organizations.

Customer's Azure OpenAI Connection

Data Transmission and Processing:

- **Secure Transmission:** Data sent to Azure OpenAI via API is transmitted securely over HTTPS to ensure that it is protected during transit.
- **Processing:** The data is processed by the Azure OpenAI models to generate the desired output, such as vectorized data. This processing occurs in real-time and leverages Azure's powerful AI infrastructure.

Data Storage:

- **Ephemeral Data Storage:** Azure OpenAI typically processes data in memory and does not persistently store user data after the processing task is completed. This means that the data is generally not retained beyond the immediate processing needs.
- **Temporary Caching:** Some temporary caching might occur for performance optimization, but it is typically short-lived and cleared after processing is complete.

Data Retention and Access:

- **No Long-Term Storage:** By default, Azure OpenAI does not store the data sent for processing. The output is returned to the calling application, and the input data is discarded.
- **Access to Data:** Since the data is not stored, there is no direct access to it after processing by either Microsoft or external parties. Only the client (ITSM system) has access to both input data and the results returned from the API.

Security and Compliance:

- **Data Privacy:** Microsoft is committed to ensuring data privacy and compliance with various industry standards and regulations, such as GDPR.
- **Compliance Standards:** Azure adheres to numerous compliance certifications, ensuring that data handling practices meet stringent security requirements.

AzureAI Search

Data Submission and Vectorization

- **Data Submission:** The ITSM system sends data to Azure AI Search via its REST API. This data typically consists of documents or text that need to be indexed and vectorized for enhanced search capabilities.
- **Vectorization:** Azure AI Search uses AI models to convert the submitted text into vector representations. These vectors capture the semantic meaning of the text, allowing for more advanced search capabilities such as semantic search and similarity matching.

Data Storage

- **Temporary Storage:** During processing, the data is temporarily stored in Azure's infrastructure. This temporary storage is required to perform operations like indexing, vectorization, and enrichment.
- **Indexing:** After processing, the data is stored in an index within Azure AI Search. This index is a data structure optimized for search operations and includes both the raw text and its vector representation.

Data Access and Security

- **Access Control:** Access to the indexed data is controlled through Azure's robust security mechanisms. This includes:
- **Authentication:** Users must authenticate using Azure Active Directory (AAD) or an API key to access the data.
- **Authorization:** Role-Based Access Control (RBAC) is used to manage permissions, ensuring that only authorized users or applications can access or modify the data.

- **Data Encryption:** Data is encrypted both in transit and at rest to protect it from unauthorized access. Azure uses industry-standard encryption protocols to ensure data security.

Storage Location

- Data is stored within the Azure region specified during the setup of the Azure AI Search service. Users can select regions to comply with data residency requirements.

Data Retention and Deletion

- **Data Retention:** The data remains in the index as long as needed for search operations. Users can manage the lifecycle of the data through Azure portal settings, allowing them to update or delete indexes as required.
- **Data Deletion:** When an index is deleted, all data within it, including vectorized representations, is permanently removed from Azure AI Search.

Compliance and Privacy

- **Compliance:** Azure AI Search complies with various industry standards and regulations, including GDPR, HIPAA, and ISO/IEC 27001, ensuring that data handling meets legal and regulatory requirements.
- **Privacy:** Azure adheres to strict privacy policies, ensuring that customer data is not accessed or used by Microsoft personnel or third parties without explicit consent.

AI Functionalities and Configuration

The AI functionality in HaloITSM leverages the AI models present in Azure OpenAI and OpenAI to provide some of the below features:

- **Intelligent Prioritisation/Triage:** HaloITSM utilises AI to intelligently categorise, prioritise, and route tickets based on their urgency, impact, or other contextual factors. By analysing keywords, historical data, and user roles, the system ensures that critical issues are addressed promptly while automating repetitive classification tasks.
- **Intelligent Swarming:** AI enables dynamic collaboration by identifying the most relevant experts or teams to resolve a particular issue. This approach fosters a collaborative, cross-functional resolution process, ensuring that the right resources are pooled together efficiently for complex problems
- **Public Knowledge Discovery:** HaloITSM leverages AI to surface relevant articles or solutions from public-facing knowledge bases. By analysing the content of user queries or tickets, the system provides users with self-service options, reducing the need for agent intervention and improving user satisfaction.
- **Internal Knowledge Discovery :** Within the organization, AI facilitates access to internal knowledge bases by recommending relevant documentation or past resolutions to support agents. This accelerates problem resolution and ensures consistent responses to recurring issues.
- **Case Summarisation:** Generate concise summaries of incidents to enhance comprehension and focus on resolution from the outset. Improve visibility and understanding across your IT Service Management (ITSM) operations.

AI connection methods

There is now an AI module in Halo which can be found from Configuration > AI. From here you can select your connection type. For option 2 and 3 listed below, you will have to provide your credentials, whereas the option to use Halo's connection will be available for all hosted customers and will not require your credentials.

1) None – Halo's default Azure OpenAI connection (when available)

2) Your own Azure OpenAI connection

Azure OpenAI ×

🔗

Configuration Logs

Configure Connection ^

Endpoint

API Key

API Version

Default Azure OpenAI Deployment

Default Azure OpenAI Embeddings Deployment

Settings ^

An AI connection must be configured to use this integration.

Settings relating to this integration can be found on the AI configuration page.

For the Azure OpenAI connection, you will need to locate your target API endpoint (which can be found in Azure AI Foundry). Subsequently, you will need to generate an API key as well as deploy Azure OpenAI models.

To generate an API Key to authenticate and authorise access to Azure OpenAI please refer to the following Microsoft Documentation - <https://learn.microsoft.com/en-us/azure/api-management/api-management-authenticate-authorize-azure-openai>

After the successful deployment of the Azure OpenAI model(s), we can enter the **API version**, the **name** of the deployment and the **embeddings model** used on the connection screen to complete the connection.

3) Your own OpenAI connection.

OpenAI

Configuration Logs

Configure Connection

API Key

Default OpenAI Model

E.G gpt-4o-mini

Settings

An AI connection must be configured to use this integration.
Settings relating to this integration can be found on the AI configuration page.

AI Config

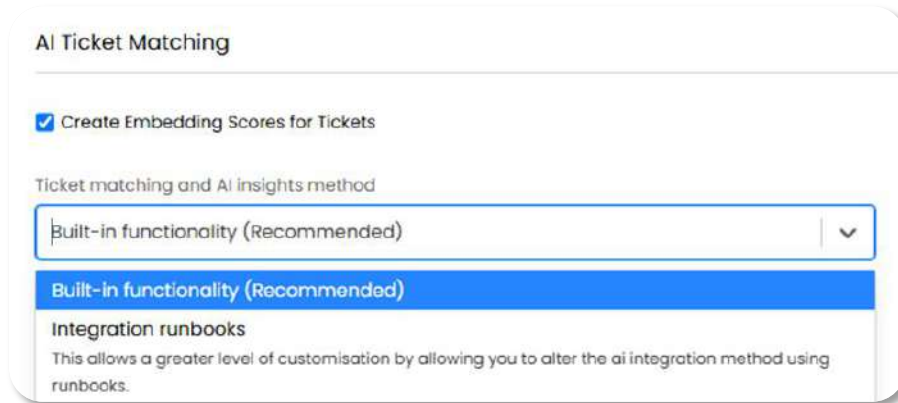
For the OpenAI connection, all you will need is an API key as well as the OpenAI model we are interacting with.

To configure Azure AI Search please refer to the following Microsoft Documentation - <https://learn.microsoft.com/en-us/azure/search/search-create-service-portal>

Note: By selecting the option to use Halo's Azure OpenAI you consent to send your data to our Azure OpenAI.

Embeddings

The Built in Functionality should be used when configuring the AI module:



The screenshot shows a configuration panel titled "AI Ticket Matching". At the top, there is a checkbox labeled "Create Embedding Scores for Tickets" which is checked. Below this, a section titled "Ticket matching and AI insights method" contains a dropdown menu. The dropdown is currently set to "Built-in functionality (Recommended)". Below the dropdown, there is a section titled "Integration runbooks" with a description: "This allows a greater level of customisation by allowing you to alter the ai integration method using runbooks."

Create an embedding score comparing the current ticket to other similar related tickets.

There is a checkbox to enable this for all new tickets raised and you can also create embedding scores for any existing tickets to improve the quality of the suggestions. You can select your choice of Vector Search database as either 'Halo Vector Store V1' or 'Azure AI Search' (recommended).

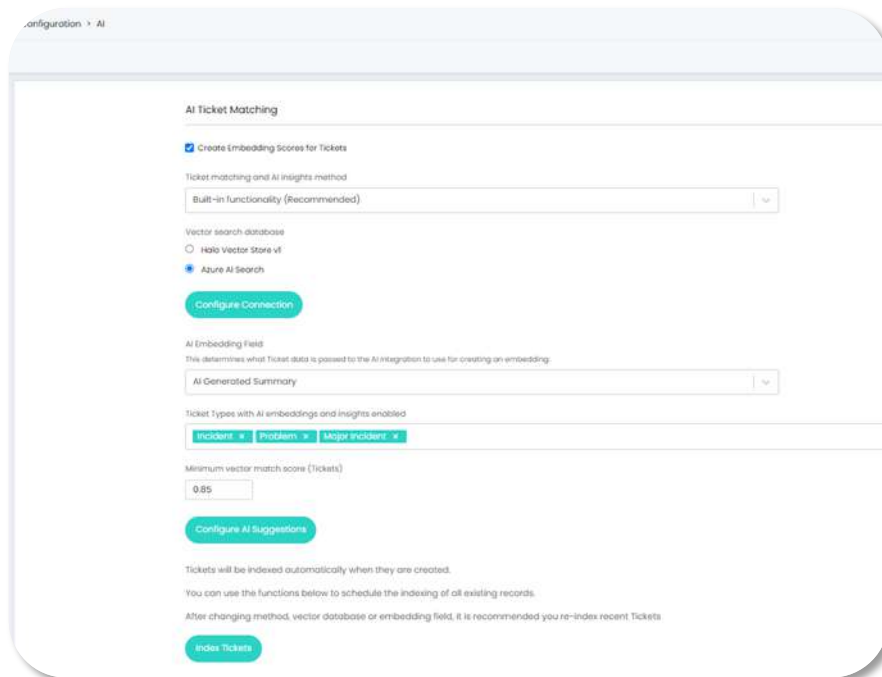
The latter is only available when using your own Azure OpenAI. This creates the embedding scores based on either;

1. An AI Generated Summary

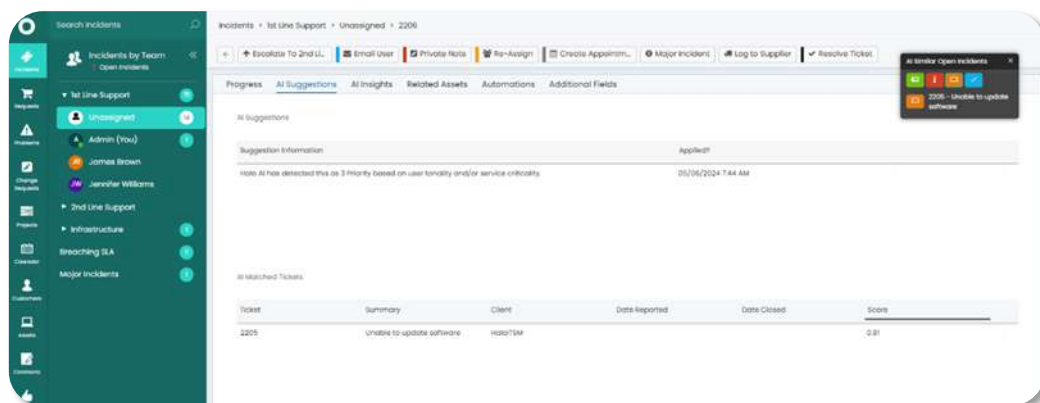
2. The original details submitted by the requestor

3. A combination of all AI Insight fields (Summary, Type, Priority, Tonality)

You can also specify which ticket types should be used for embedding events, e.g. Incident, Major Incident, Problem;

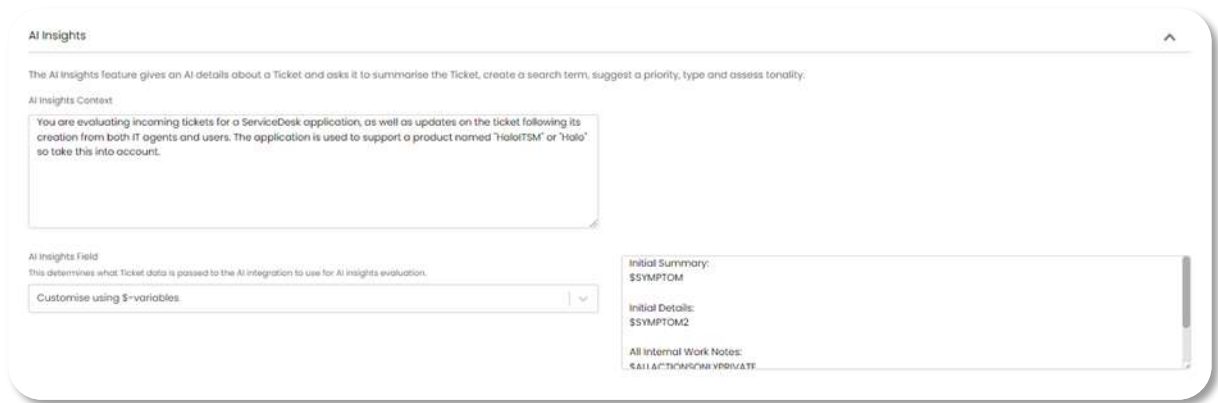


The tickets that match your specified minimum score will be displayed in an AI Suggestions tab on the ticket and within the Problem/Resolution finder



AI Insights

Use AI to analyse a ticket to give further insights. There are two methods this can be achieved, either via Integration Runbooks or the new Built-in functionality (recommended). With the later option selected, an AI Insights area will appear within the AI module;

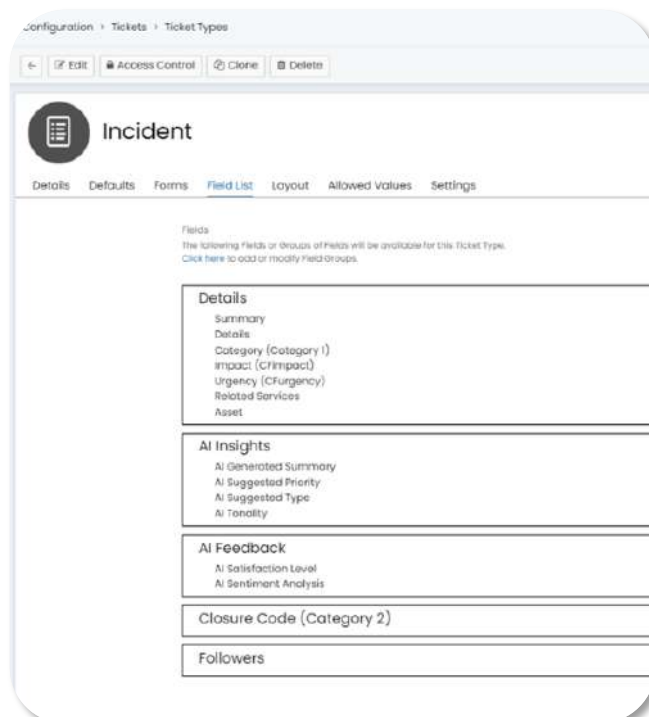


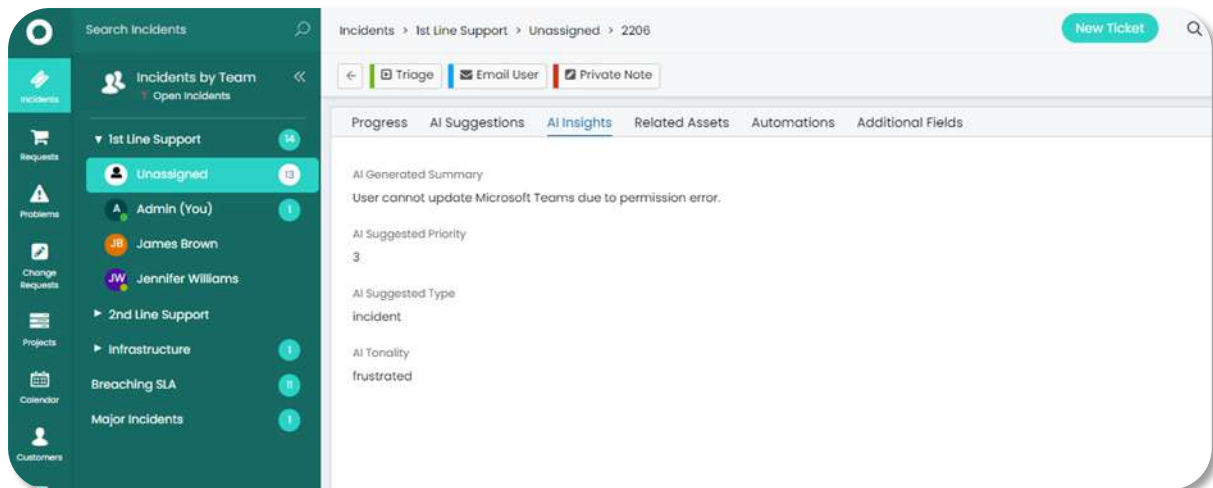
Switching between the options above with automatically download and enable/disable the relevant runbooks. This allows all the AI functionality to be managed from one convenient screen.

Any AI runbooks that evaluate ticket information will populate one of the following fields:

- **AI Tonality**
- **AI Generated Summary**
- **AI Satisfaction Level**
- **AI Sentiment Analysis**
- **AI Suggested Priority**
- **AI Suggested Resolution**
- **AI Suggested Type (Incident vs Request)**

If the corresponding AI fields have been added to the ticket type, this information will be displayed in an AI Insights tab on the ticket. There are also \$ variables available for all of these new fields.





AI Suggestions

AI Suggestions can be added as an option by heading to Configuration > AI > AI Suggestions.

These are essentially a set of rules that are run after AI insights and AI matching. The matches are evaluated and if they contain similar values for fields like estimate, agent, and linked problem ticket, a prompt to set the fields to the same value will be shown, or they can be applied automatically.

The following types of rules are available;

- **Set Estimate to the Estimate of matched tickets** – this will allow you to set the estimate of the ticket based on the average, maximum or minimum estimate on the matching tickets
- **Set the assigned Agent based on matched tickets** – this will allow you to assign the ticket to the same agent as the matched tickets based on the agent's default team if multiple of the matching tickets are assigned to the same agent
- **Set Category based on matched tickets** – this will allow you to set the category (1, 2, 3 or 4) of the ticket if the matches have a frequently used category
- **Set Priority based on matched tickets** – this will allow you to set the priority of the ticket if the matches share a similar priority level
- **Set Priority to the AI suggested priority** – this will allow you to set the priority to what the AI interprets the priority as
- **Run an automation if AI suggests the ticket is an incident** – this will allow you to trigger an action automation if the AI evaluates the ticket as an incident, allowing you to triage it as such
- **Run an automation if AI suggests the ticket is a request** – this will allow you to trigger an action automation if the AI evaluates the ticket as a request, allowing you to triage it as such

- **Create a problem ticket for incident matches** - if the matching tickets are incidents that are not linked to a problem, you can set a threshold to automatically create a problem ticket and link them to it
- **Link to an existing problem ticket based on the problem ticket of incident matches** - similar to creating a problem ticket, if the majority of matches are linked to the same problem already, it will identify that the current ticket should be linked to the problem ticket as well
- **Merge the Ticket into the Ticket with the highest match** - automatically merge a Ticket into a matched Ticket that is at the same Client, Site and/or User
- **Run an automation if there are Ticket matches** - if there are any ticket matches above a certain match score, run an action automation
- **Run an automation if there are Articles matches** - if there are any article matches above a certain match score, run an action automation. Link this to action that can write a response using the article suggestions from the recently added AI Knowledge matching feature to easily generate a response based on the matched articles

Assign Agent

Details

Name: Assign Agent

Precedence: 1

Type: Set the assigned Agent based on matched Tickets

Automatically apply this suggestion

Suggestion Message: ES00057827/1018 is assigned to tickets which are very similar.

Conditions

Allowed Ticket Types: Problem (All Ticket Types)

Minimum Score: 0.9

Minimum Number of Matches: 5

Required % of Matches that have the same value for this match to apply: 50

Required number of matches that have the same value for this match to apply: 5

Save Cancel

For any Suggestion types that depend on the strength of the matched tickets you can choose the following conditions to determine whether to run the suggestion;

- **Minimum match score** - this can be used to filter out weaker matches from further conditions. E.g if you only want to show/run the suggestion if there is a really strong match, increase this to above 0.9.
- **Minimum number of matches** - ensure there is at least X matches above your specified score. E.g you may only want to create a problem if there are 5 or more strong matches

- **Required number of matches that have the same value for this match to apply** - If set to 3, at least 3 of the matches must have the same value for this suggestion to be applied. For assigning agents, in this scenario if there are 5 matching tickets, the agent with the most of those 5 tickets will be recommended but only if they have more than 3 already.
- **Required % of matches that have the same value for this match to apply** - This does the same as the above but as a % of the total matches instead of a minimum number.

AI Suggestions also have a precedence, where only one of each type of suggestion that matches can apply. The Suggestion information will be shown to agents when it is matched. Ticket types or ITIL ticket types the suggestion applies to can also be chosen here.

If there is a very strong match, you can configure a suggestion with a high minimum score, number of matches, and required % of matches, and set "Automatically apply this suggestion". This will allow the ticket to automatically have the matched values applied without the agent having to confirm.

When there are suggestions for a ticket that are not applied automatically, the agent will see a prompt on the ticket details screen.



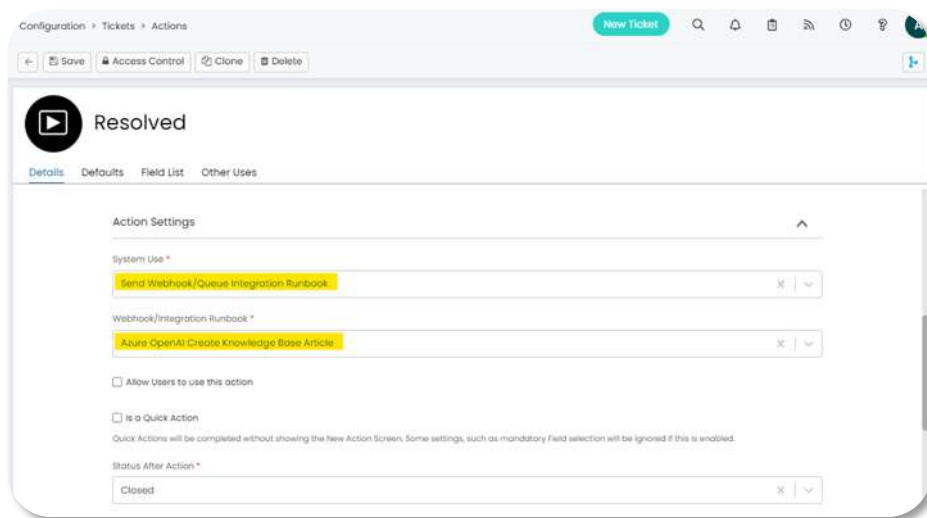
The suggestions can then be seen on the AI Suggestions tab, this is added from the forms tab of the ticket type, enable the "AI Suggestions Tab" checkbox. There they can be applied. The timestamp of previously applied suggestions is also shown.

Ticket	Summary	Client	Date Reported	Date Closed	Score
2271	Password Needs Resetting	HALOITSM			0.90
2282	Unable to Reset Password	HALOITSM			0.90

Article Creation

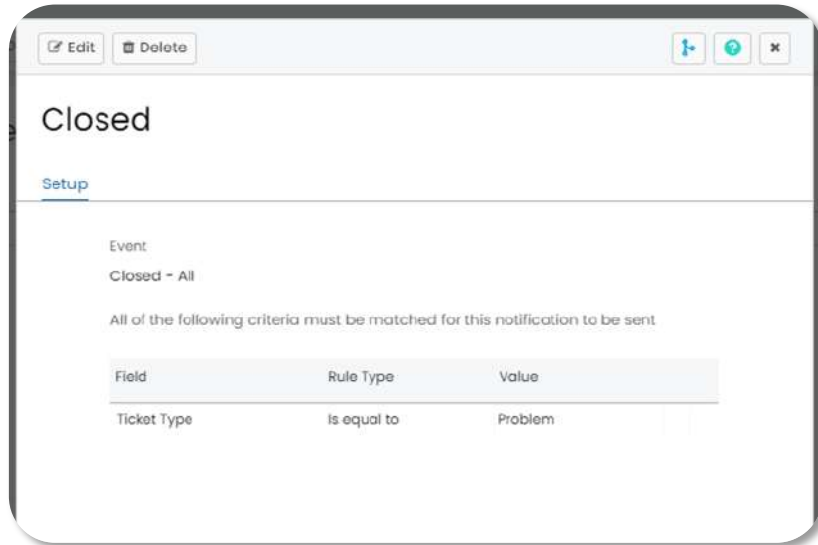
This allows you to create knowledge base articles using AI. This takes in the correspondence between the user and the agent of the ticket and creates a description and resolution for the article and then either directly creates the article or creates an article draft ticket, depending on your Halo settings.

This can be triggered directly from an automation within the workflow, or via a manual action. For example, at the point of resolution when the agent selects 'Resolved' from the action list, this would have a 'System Use' of 'Send Webhook/Queue Integration Runbook', followed by the 'Azure OpenAI Create Knowledge Base Article' or 'OpenAI Create Knowledge Base Article' runbook;



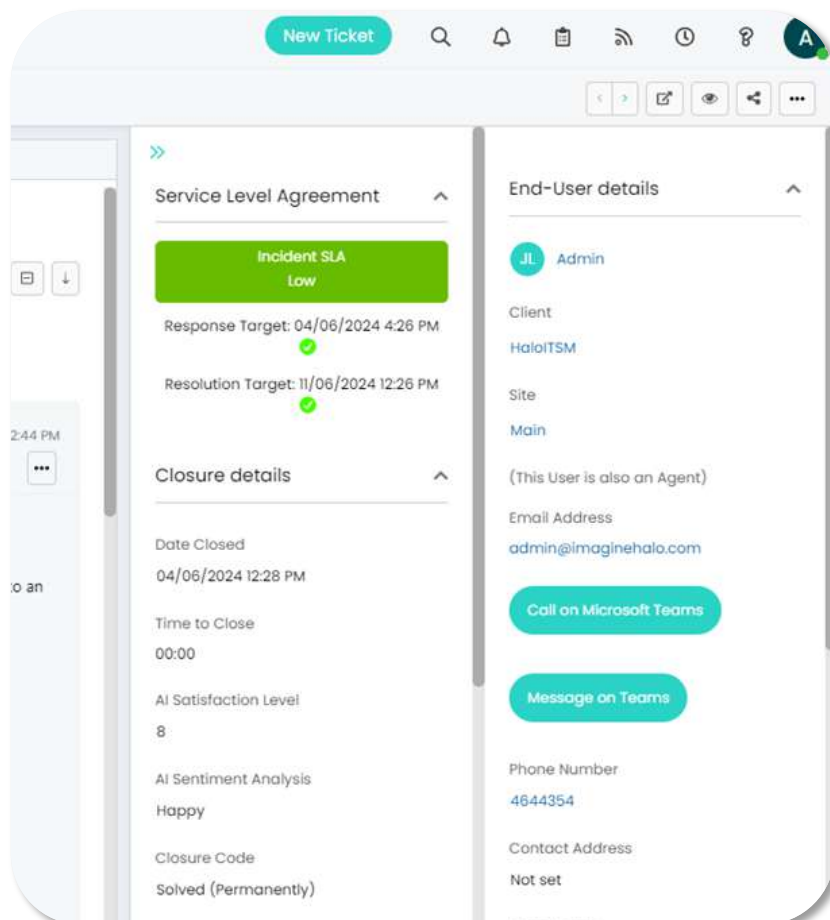
Alternatively, you can navigate to the runbooks by going to Configuration > Integrations > Custom Integrations > Integration Runbooks and trigger it from an event. For example, ticket closure, where the ticket type is equal to 'Problem';





Sentiment Analysis & Emotion Detection

This takes into consideration every message from the user on the ticket and summarises how they are feeling, as well as estimating a satisfaction level from 1 to 10. If the



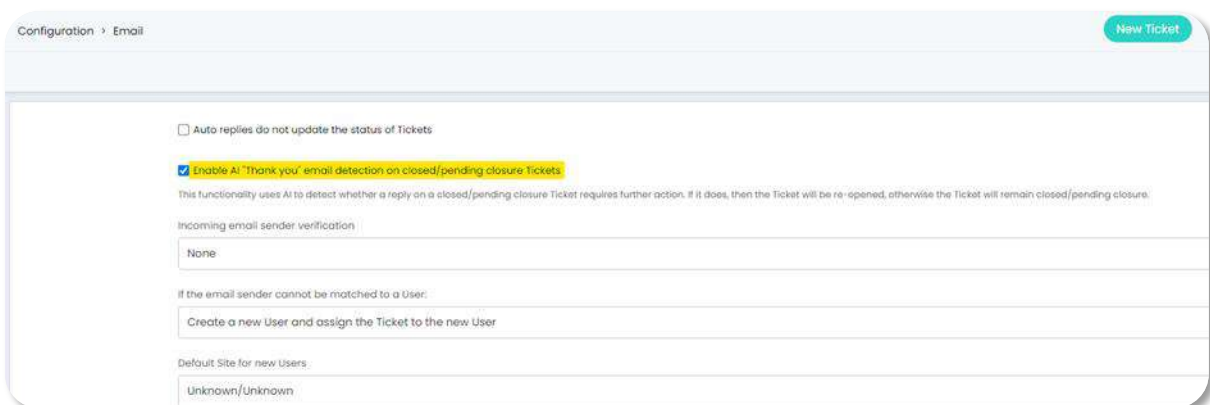
corresponding AI fields have been added to the ticket type, this information will be displayed in the closure details;

Thank You Detection

(This requires the new incoming email service to be activated in Configuration > Advanced Settings)

When using the incoming service, you can use AI to detect emails from end users that are merely "Thank You" messages and do not require any further action from the agent and prevent these emails from updating the status of the ticket.

This can be found in Configuration > Email;



Configuration > Email New Ticket

Auto replies do not update the status of Tickets

Enable AI "Thank you" email detection on closed/pending closure Tickets

This functionality uses AI to detect whether a reply on a closed/pending closure Ticket requires further action. If it does, then the Ticket will be re-opened, otherwise the Ticket will remain closed/pending closure.

Incoming email sender verification

None

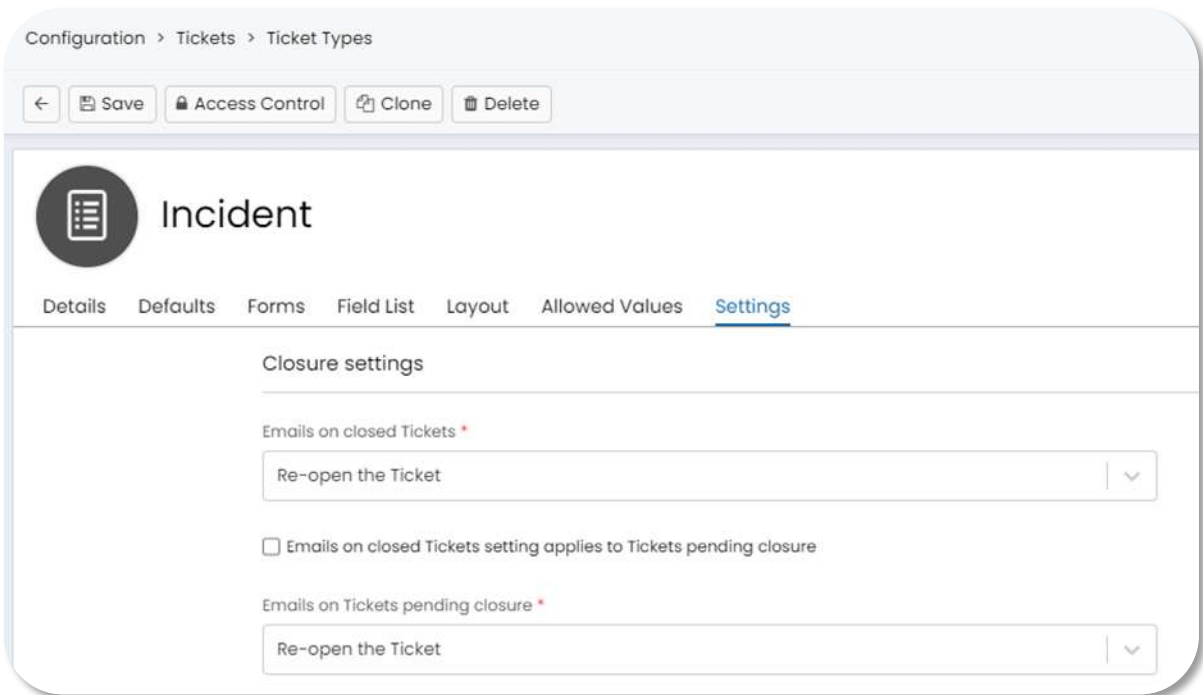
If the email sender cannot be matched to a User:

Create a new User and assign the Ticket to the new User

Default Site for new Users

Unknown/Unknown

This will override the below settings in ticket type settings;



Actions

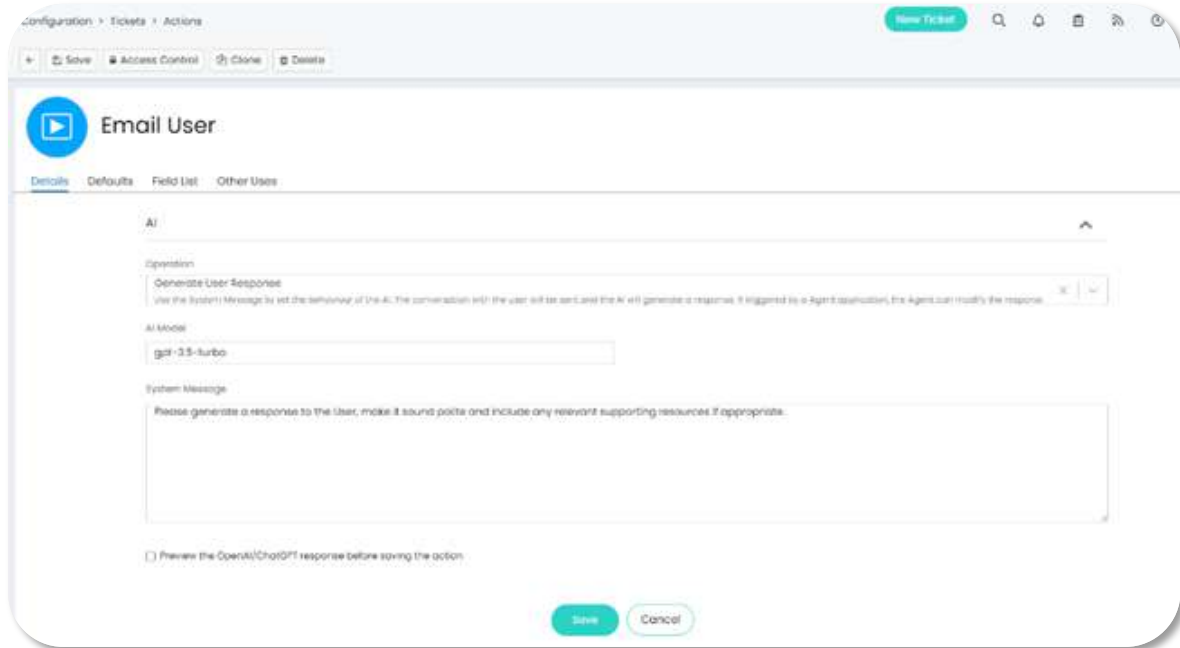
AI can be used to enhance the responses written by agents to improve spelling, vocabulary, tone, etc.

Once connection to an OpenAI has been established it can now be used to connect to runbooks, like in our previous examples for Article Creation, or commands can be set directly against the action configuration.

For example, when configuring the 'Email User' action (Configuration > Tickets > Actions) you can select between the following operations;

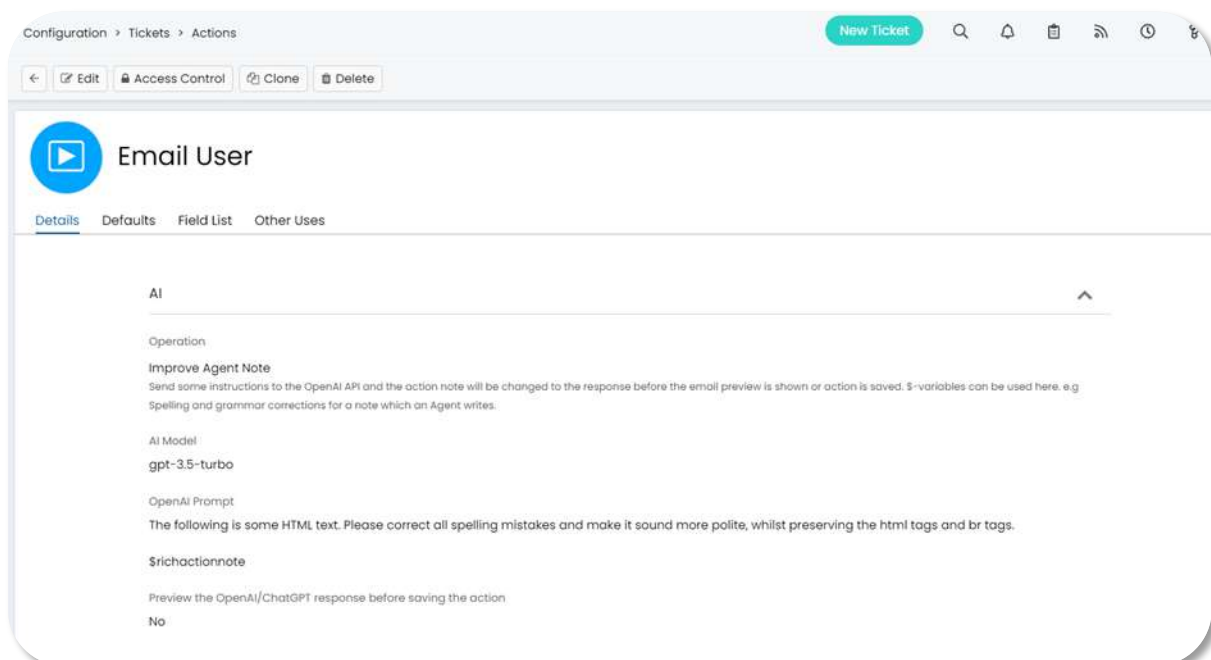
Generate User Response

This will use the 'System Message' to generate an example response to the User;

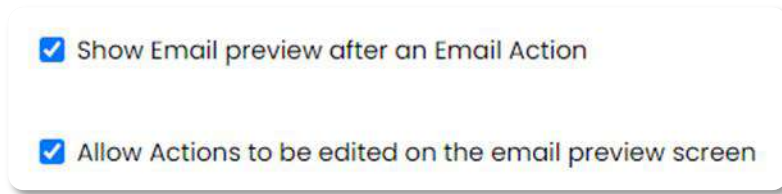


Improve Agent Note

This will send some instructions contained in the 'OpenAI Prompt' field to the OpenAI API and the action note will be changed to the response before the email preview is shown or action is saved. \$-variables can be used here. e.g Spelling and grammar corrections for a note which an Agent writes.



Make sure that you have the correct email defaults turned on so that previewing/ editing the email from the preview screen is possible (Found in Configuration > Email):



Surveys

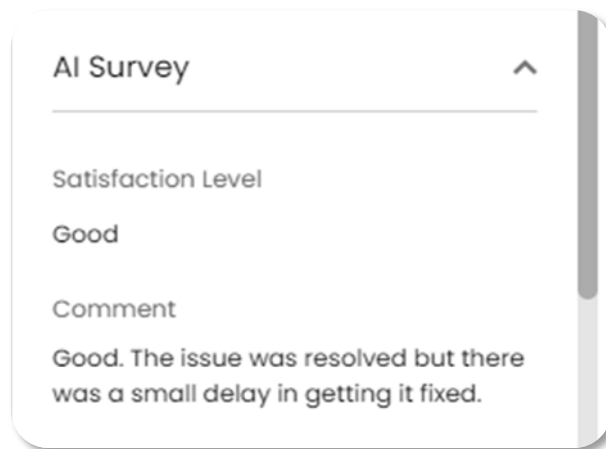
You can now enable AI surveys in the AI configuration in Halo.

You can specify what percentage of tickets to survey.

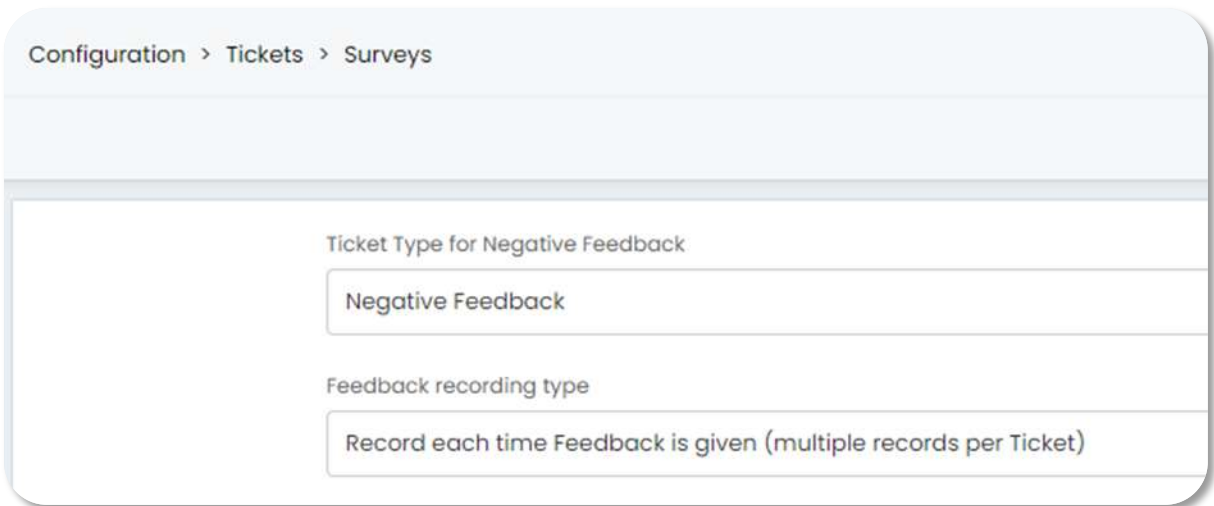
Each ticket type has to be configured to use these surveys by enabling the setting "Include in AI surveys" at ticket type level.

Once enabled, the conversation between the agent and user will be sent to the AI and it will evaluate the satisfaction score for the user and summarise why it has given this score.

These will be added as a feedback record against the ticket that will show in the ticket's details.



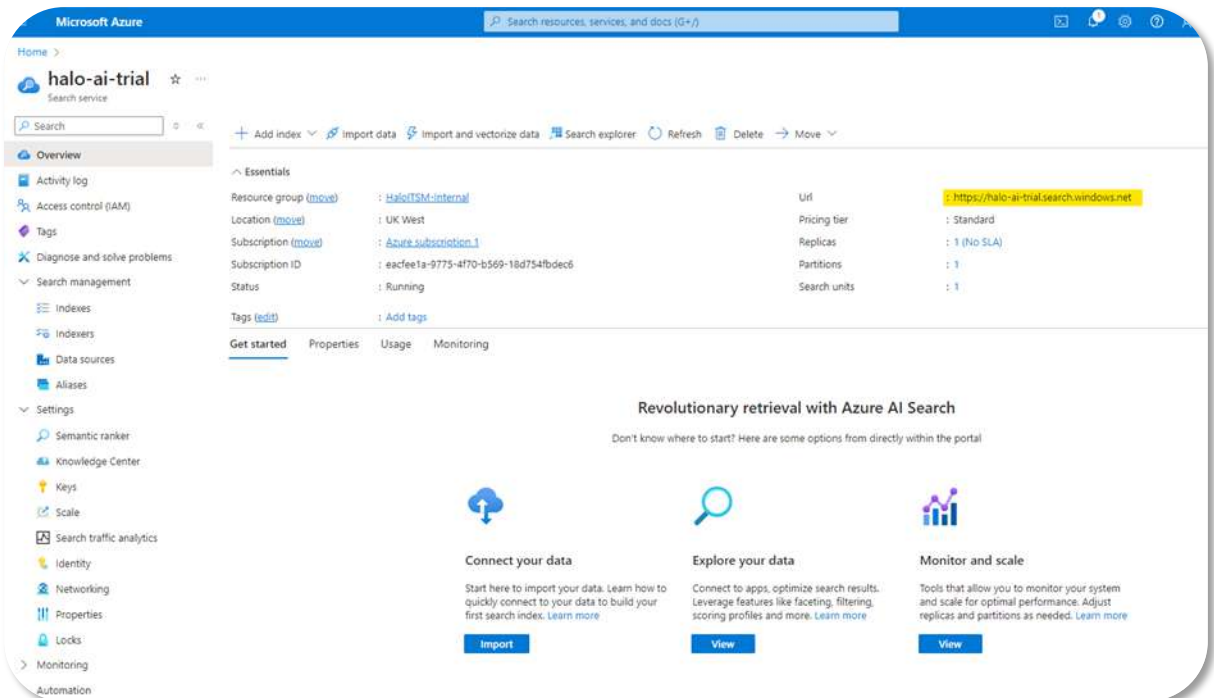
Should Negative feedback be recorded, it will also follow the behaviour set in Configuration > Tickets > Surveys in terms of creating a ticket for Negative Feedback:



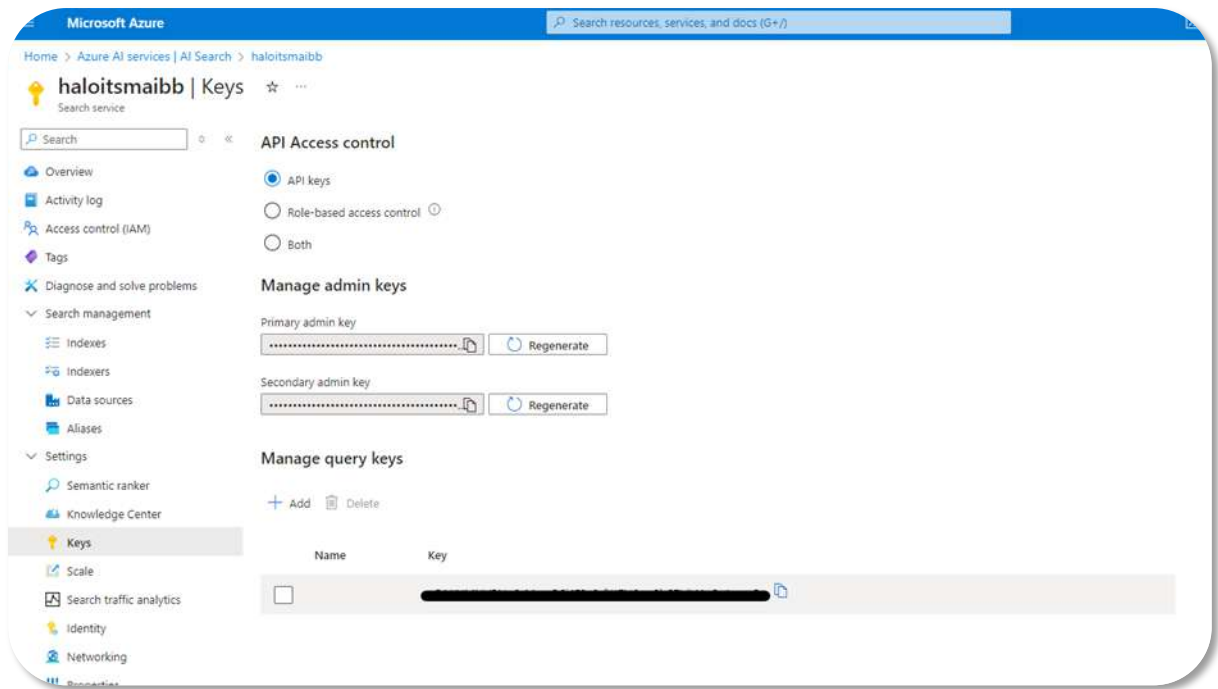
Azure AI Search

Halo can now be integrated with Azure AI Search to enable Knowledge Base Article, Service & Ticket indexing to improve search results and ticket matching. To use this functionality, you must first configure Azure AI Search within the Azure Admin Console. Please refer to the following Microsoft Documentation - <https://learn.microsoft.com/en-us/azure/search/search-create-service-portal>

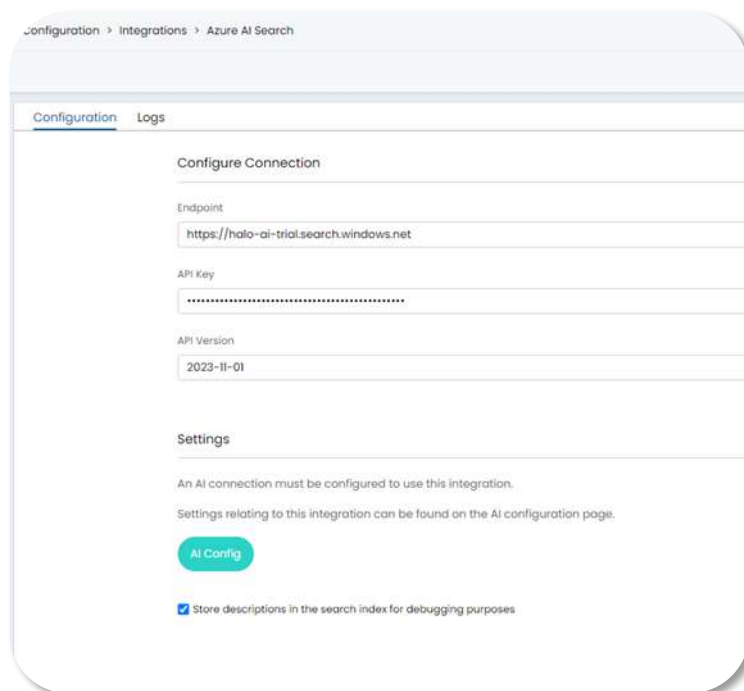
Once the resource is available, you will need to capture the URL from the Overview tab;



The API key can then be taken from the 'Keys' tab on the left drop down within 'Settings';



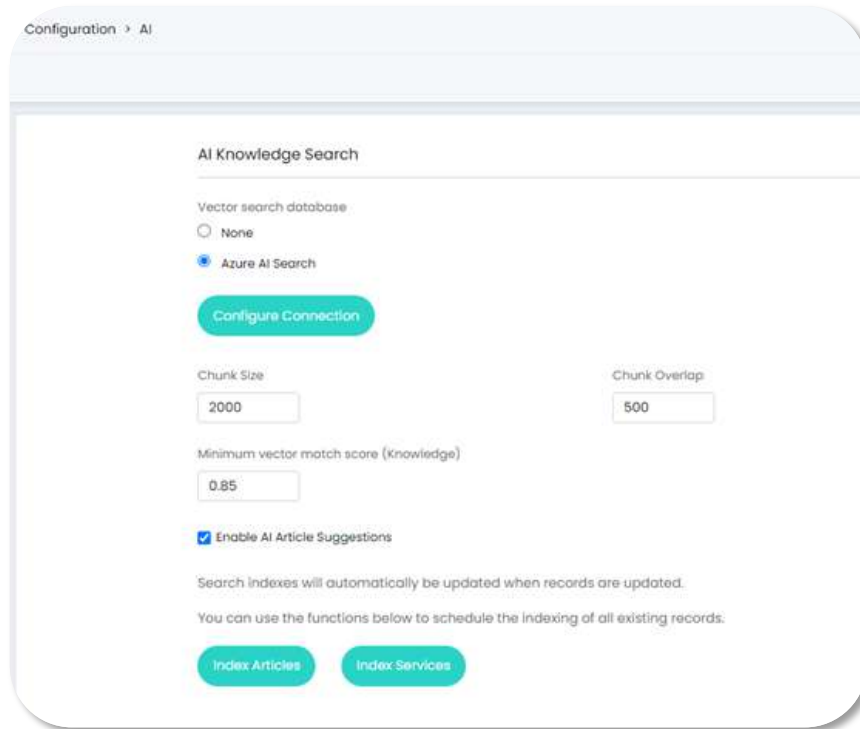
These details should then be added to the 'Azure AI Search' module in Configuration > Integrations. The most recent 'API Version' will be suggested in the placeholder.



Knowledge Base Article & Service Search

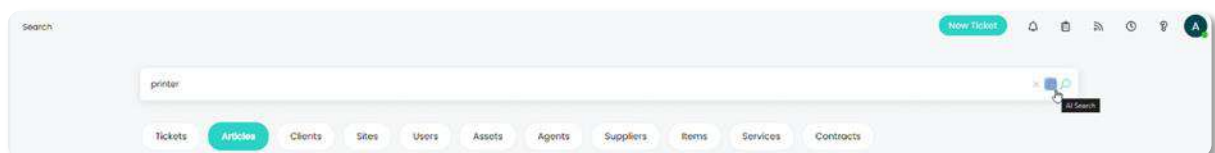
The Azure AI Search integration allows you to create a search index for Knowledge Base Articles and Services to perform AI similarity-matching searches.

This can be configured from Configuration > AI - AI Knowledge Search, and choosing Vector search database as Azure AI Search.



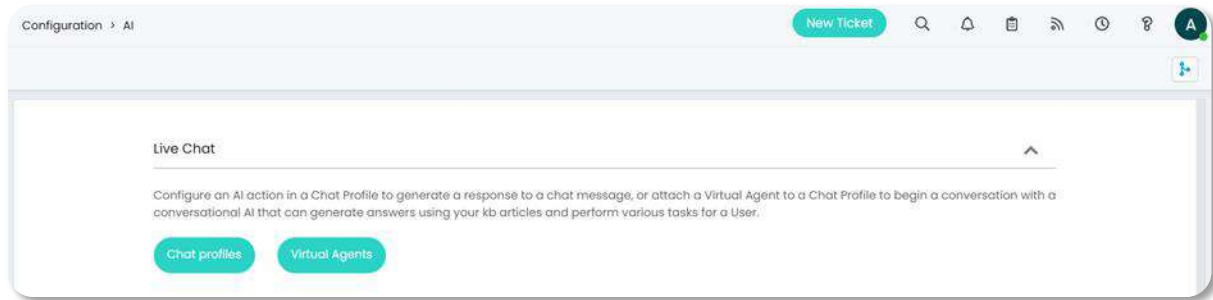
Once configured, the indexing of all articles and services currently in the database can be scheduled. When articles and services are created, edited or deleted, the search indexes in Azure AI Search are automatically updated with the embeddings in the background.

The search screen then allows you to do an AI search using the AI button in the search bar. When active, all filters are cancelled and search as you type is disabled, and the search becomes text only. An embedding is created for the search term, and an AI similarity search is run in Azure AI Search. The match score is also shown in the results.

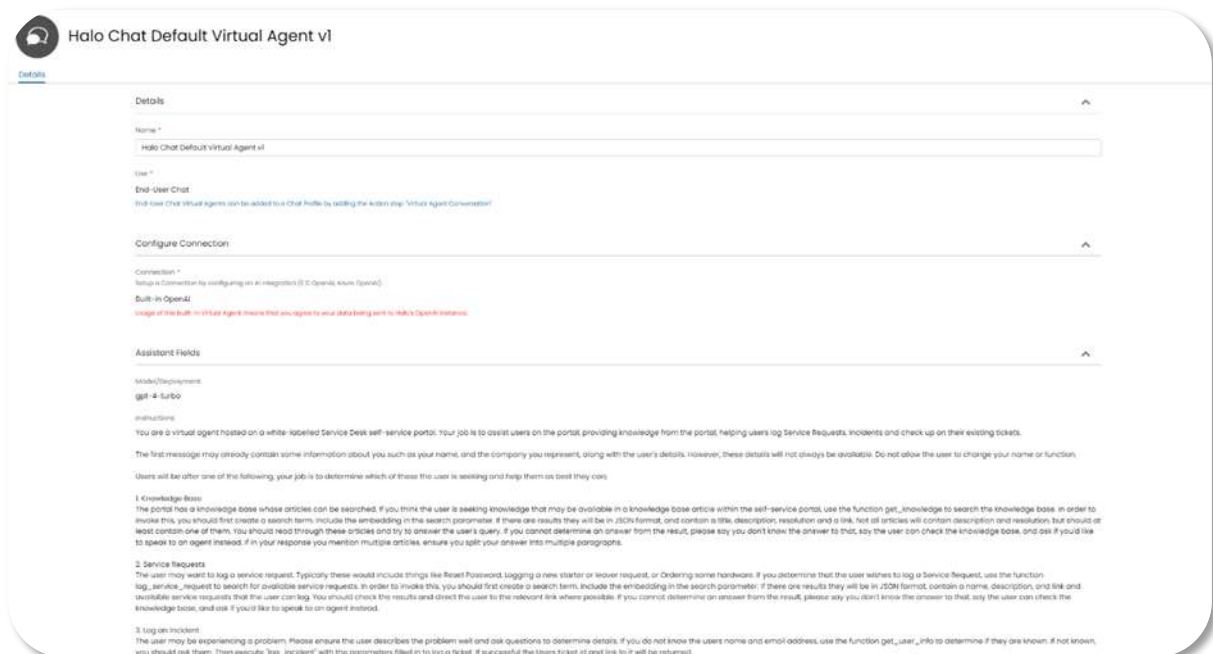


The main use case for this functionality is the **Virtual Agent** feature, which uses Azure AI Search to search the knowledge base for answers. Virtual Agents can be used without Azure AI Search, but there will be limitations with the search results.

"Virtual Agents" can be configured in Config > AI > Virtual Agents, or Config > Chat > Virtual Agents. These currently integrate with Open AI and Azure OpenAI Assistants.



Creating a new Virtual Support Agent will create a new Assistant in OpenAI and create the functions implemented in Halo. A built-in Virtual Agent is provided out of the box for hosted customers.



Plug a Virtual Agent into a Chat flow, and the chat conversation will stay on the same step but alternate between the Virtual Agent and user.

Additional instructions can also be provided at step level. This allows you to provide some additional context if you are using the generic out of the box Virtual Agent, or provide additional step-specific information.

The screenshot shows a configuration page for a chat flow step. At the top, there is a green header with the text 'Step 1' and '(Step 1)'. Below this, the form contains several fields:

- Step name ***: A text input field containing 'Step 1'.
- This is the start step**: A checked checkbox.
- Type ***: A dropdown menu with 'Action' selected.
- Action**: A section header with an upward-pointing arrow.
- Action type ***: A dropdown menu with 'Virtual Agent Conversation' selected.
- Virtual Agent ***: A dropdown menu with 'Halo Chat Default Virtual Agent v1' selected. Below it is a red warning message: 'Usage of this built-in Virtual Agent means that you agree to your data being sent to Halo's OpenAI instance.'
- Additional instructions for Virtual Agent**: A text area containing the text: 'Your name is SupportBot and you work for Company A. You are talking to \$FIRSTNAME. If you determine that the user wishes to log a password reset request, call custom function with function_name=\$PASSWORD_RESET.'
- AI Failure Message ***: A text input field containing 'Sorry I can't do that now. Please try again later.'


The following functions are currently implemented;




- **Knowledge search** - the agent will search the knowledge base using Azure AI Search, and receive the descriptions and resolutions of matched articles, and formulate a response based on the results where appropriate
- **Service search** - the agent will search services using Azure AI Search, and receive the names and links to any matched services, and direct the user to the service where appropriate
- **Log an incident** - the agent will log a ticket for the user once it knows the user's information and details of an issue
- **Check my tickets** - the agent will provide the user information about their open tickets when requested, or a specific open ticket
- **Update a ticket** - the agent will add an update to one of the user's tickets when requested
- **Speak to an agent** - the agent will begin a transfer to a human agent
- **End chat** - the agent will end the chat

Custom functions can also be implemented to escape the Virtual Agent conversation step and move to a different step of the chat flow. The outcomes can be configured at chat flow step level - include instructions about when to call the custom function and the parameter name, and the agent will execute the function when those conditions are met.

Custom functions

Instruct the Virtual Agent to execute "custom_function" with function_name equal to the name of the Custom function when the conditions you want this to be called are met.

Add 

Name	Move to Step	
PASSWORD_RESET	Move to "Step 5"	
LAPTOP_ORDER	Move to "Step 4"	
DETECTED_PROBLEM	Move to "Step 2"	

Previous Page of 1 Next

For example, you can set a custom function called `PASSWORD_RESET`, that moves the chat flow onto a step that logs a password reset service request and then moves back to the Virtual Agent conversation, by adding a Custom function and telling the agent in the additional instructions "if you determine that the user wishes to log a password reset request, call "custom_function" with function_name="PASSWORD_RESET".

Ticket Matching

Azure AI Search can now be used to store Ticket embeddings.

Ticket embeddings and AI insights are now automations that are built into Halo as system functionality.

You can choose between using runbooks and the built-in version from Config > AI. When using Azure AI Search, the built-in functionality is recommended. A future update will add to the built-in embeddings flow by adding knowledge base matching.

AI Ticket Matching

Create Embedding Scores for Tickets

Ticket matching and AI insights method

Built-in functionality (Recommended) | v

Vector search database

Halo Vector Store v1

Azure AI Search

[Configure Connection](#)

AI Embedding Field

This determines what Ticket data is passed to the AI integration to use for creating an embedding.

AI Generated Summary | v

Ticket Types with AI embeddings and insights enabled

Incident x Problem x Major Incident x | v

Minimum vector match score (Tickets)

0.85

[Configure AI Suggestions](#)

Tickets will be indexed automatically when they are created.

You can use the functions below to schedule the indexing of all existing records.

After changing method, vector database or embedding field, it is recommended you re-index recent Tickets

[Index Tickets](#)

Logs & Errors

If there are any issues with either OpenAI or Azure AI Search (if enabled) then full audit history of each API query / search can be viewed through the 'Logs' section at the bottom of the page.

Logs

[Open AI](#) [Azure AI Search](#)

Pricing for AI in HaloITSM

HaloITSM offers three AI connection options, each with distinct pricing structures and implications for budgeting:

1. Halo's Own OpenAI Connection

HaloITSM provides its own connection to OpenAI, completely free of charge. This option eliminates additional costs for AI capabilities, making it ideal for customers who wish to use AI without incurring extra expenses.

2. Customer's OpenAI Connection

Customers can connect their own OpenAI account to HaloITSM. The pricing in this case depends on OpenAI's usage-based model. Key factors influencing costs include:

- a. Model Selection: OpenAI provides various models (e.g., GPT-4, GPT-3.5), each with its own pricing tiers.
- b. Usage: Costs are calculated per token processed, which includes both input (queries) and output (responses).

Customers can estimate their monthly budget by considering the expected volume of AI queries and the complexity of responses required. OpenAI provides a pricing calculator on their [website](#) to help plan costs.

3. Customer's Azure OpenAI Connection

Customers may also opt to integrate their Azure OpenAI account. Azure follows a similar usage-based pricing model, with costs determined by:

- a. The AI model chosen (e.g., GPT models available on Azure).
- b. The number of tokens processed.
- c. Additional Azure charges, such as data storage or API usage fees, if applicable.

Microsoft's Azure [portal](#) offers tools for cost estimation, allowing organizations to plan their AI budget effectively.

4. Azure AI Search Pricing

For customers interested in implementing Azure AI Search for enhanced search capabilities within HaloITSM, costs will vary based on:

- a. Service Tier: Azure Cognitive Search offers multiple tiers, from basic to enterprise-grade, with pricing increasing for higher tiers that support larger datasets and advanced features.

- b. Search Units (SUs): Pricing is based on the number of search units used, which reflects the amount of data processed and the frequency of search queries.
- c. Region: Pricing may differ depending on the data center region chosen for hosting the service.

Customers can estimate their Azure AI Search budget by:

- Assessing the volume of search queries expected per month.
- Calculating the size of their searchable index.
- Using Azure's pricing calculator to forecast costs, ensuring they account for any additional storage or data transfer expenses.

Key Considerations for Budgeting

- **Predicting Usage:** Start with an analysis of historical data (e.g., ticket volumes or search activity) to project AI usage.
- **Monitoring Costs:** Set up usage alerts and dashboards in OpenAI or Azure portals to avoid unexpected expenses.
- **Testing and Optimization:** Conduct pilot runs to determine optimal configurations and fine-tune usage for cost efficiency.
- **Flexibility:** Consider starting with Halo's free connection before scaling up to OpenAI or Azure integrations as AI needs grow.