Some products and features are in the process of being renamed. Generative playbook and flow features are also being migrated to a single consolidated console. See the details (/dialogflow/cx/docs#consolidation).

# Agent settings

Agents have many settings that affect behavior. Each console provides different settings.

## Conversational Agents console agent settings

To access agent settings:

Console (#console) API (#api)

1. Open the Conversational Agents console (https://conversational-agents.cloud.google.com).
2. Choose your Google Cloud project.
3. Select your agent.
4. Click these settings icon in the top right corner.
5. Update the settings as desired.
6. Click **Save**.

The following subsections describe the different categories of agent settings.

### General settings

The following general settings are available for agents:

- **Display name** (#settings-general-display-name)

  A human-readable name for your agent.

- **Time zone** (#time-zone)

  The default time zone for your agent.

- **Conversation start** (#conversation-start)

  This setting determines whether your agent starts the conversation with a generative playbook or a deterministic flow.

- **Agent location** (#settings-general-location)

  The agent region you selected when creating the agent.

- **Agent lock** (#agent-lock)

  - **Lock the agent**

    Indicates whether the agent is locked. A locked agent cannot be edited.

- **Logging settings** (#logging-settings)

  - **Enable Cloud Logging**

    Indicates whether Cloud logging (/dialogflow/cx/docs/concept/logging) is enabled for the agent.

  - **Enable conversation history**

    Indicates whether you would like Google to collect and store redacted (/dialogflow/cx/docs/concept/security-settings) end-user queries for quality improvement.

    ★ **Note:** Some features, such as Experiments (/dialogflow/cx/docs/concept/experiments), Conversation history (/dialogflow/cx/docs/concept/conversation-history), BigQuery export (/dialogflow/cx/docs/concept/export-bq), Analytics (/dialogflow/cx/docs/concept/analytics), and NLU model improvement require this setting to be enabled, as they rely on traffic logs.

  - **Enable consent-based end-user input redaction**

    If this setting is enabled, it lets you use a special session parameter to control whether end-user input and parameters are redacted from conversation history (/dialogflow/cx/docs/concept/conversation-history) and Cloud logging (/dialogflow/cx/docs/concept/logging). By default the session parameter is `true`. If this setting is disabled, no redaction occurs.

    User consent is collected using a boolean session parameter: `$session.params.conversation-redaction`. If this setting is enabled, and the session parameter is set to `false`, no redaction occurs **(other redaction strategies still apply)**. If this setting is enabled, and the session parameter is set to `true`, redaction occurs.

An example consent requesting flow could be: first ask the user if they would like to keep end-user input, and match the response with two intents, one is "yes intent" and the other is "no intent". Then, set the session parameter to `false` (no redaction) in the parameter presets (/dialogflow/cx/docs/concept/fulfillment#param-preset) of the "yes intent" route in fulfillment (/dialogflow/cx/docs/concept/fulfillment), and to `true` (redaction occurs) in the parameter preset (/dialogflow/cx/docs/concept/fulfillment#param-preset) of the "no intent" route.

- **BigQuery export** (#bigquery-export)
  - **Enable BigQuery export**

    Indicates whether BigQuery export (/dialogflow/cx/docs/concept/export-bq) is enabled.
  - **BigQuery dataset**

    The BigQuery dataset name.
  - **BigQuery table**

    The BigQuery table name.
- **Intent Suggestions** (#intent-suggestions)

  You can enable intent suggestions (/dialogflow/cx/docs/concept/intent#suggestions).
- **User feedback** (#user-feedback)

  You can enable this setting to provide feedback about the quality of the agent's answers, which includes thumbs up/down.
- **Git integration** (#git-integration)

  Use this setting to add a Git integration (/dialogflow/cx/docs/concept/github).

## Generative AI settings

The following generative AI settings are available:

- General
  - **Generative model selection** (#settings-generative-model-selection)

    Select the model used by generative features. For more information, see model versions (/vertex-ai/generative-ai/docs/learn/model-versioning).
  - **Token limits**

- **Input token limit** (#generative-input-token-limit)

  Select the input token limit for the generative model (/vertex-ai/generative-ai/docs/learn/models). This is the maximum token size for input sent to the model. Depending on the model, a token (/vertex-ai/generative-ai/docs/multimodal/get-token-count) can be somewhere between one character and one word. Smaller token limits have lower latency, but the model input size is limited. Larger token limits have higher latency, but the model input size can be larger.

- **Output token limit** (#generative-output-token-limit)

  Select the output token limit for the generative model (/vertex-ai/generative-ai/docs/learn/models). This is the maximum token size for output received from the model. Depending on the model, a token (/vertex-ai/generative-ai/docs/multimodal/get-token-count) can be somewhere between one character and one word. Smaller token limits have lower latency, but the model output size is limited. Larger token limits have higher latency, but the model output size can be larger.

- **Temperature** (#generative-temperature)

  The temperature for a LLM lets you control how creative the responses are. A low value provides more predictable responses. A high value provides more creative or random responses.

- **Context token limits** (#generative-context-token-limits)

★ **Note:** This feature is restricted access.

The setting culls some past turns from the LLM prompt in order to keep the prompt size from growing with every sequential turn handled by the agent. This feature offers a way to mitigate unwanted prompt size growth.

Normally, without truncation, each subsequent turn will be appended into the "conversation history" of the LLM prompt regardless of whether it is relevant to the current turn. This can ultimately lead to the prompt increasing in size with every turn. As more of the prompt is taken up by conversation history, less of the prompt can be used for few-shot examples (so these might get dropped). Eventually, the prompt might also breach current token limits. You can increase token sizes to accommodate this, but keep in mind that increased prompt sizes also add to the LLM response latency.

Context truncation lets you set a percentage of the token budget to be reserved for conversation history, as a maximum. Conversation turns are preserved in most recent to least recent order. This setting can help you prevent token limits from being exceeded. Regardless of which setting you choose, a minimum of two conversation turns are preserved, in most recent to least recent order.

You must first set a token limit before you can modify this setting.

Truncating context might cause some parameters to be inadvertently be lost if they are part of culled turns. Evaluate your playbook interactions carefully after enabling this option.

Token input budget is also used by the following:

- System instructions and examples: Automatically added to the prompt. This behavior cannot be modified.
- Playbook instructions and goals: Any instructions and goals that you write will be added to the prompt in their entirety.
- Playbook few-shot examples: Are added either in order (by default) or by an algorithm that you choose (such as regular expression best match ordering). Examples are culled to fit within input token budget after all other items are included.
- Conversation history made up of user and agent utterances, flow and playbook transition context, tool calls and outputs in the same session from all previous turns sequentially handled by the current playbook.

- **Banned phrases** (#generative-banned)

  List of phrases that are banned for generative AI. If a banned phrase appears in the prompt or the generated response, the generation will fail. Be aware that the list of banned phrases has consequences on every query and LLM call and will increase latency.

- **Safety filters** (#generative-safety)

  Configure sensitivity levels of safety filters with respect to different Responsible AI (RAI) categories. Content will be assessed against the following four categories:

| Category | Description |
|---|---|
| Hate speech | Negative or harmful comments targeting identity and/or protected attributes. |
| Dangerous content | Promotes or enables access to harmful goods, services, and/or activities |

| Category | Description |
|---|---|
| Sexually explicit content | Contains references to sexual acts and/or other lewd content |
| Harassment | Malicious, intimidating, bullying, and/or abusive comments targeting another individual |

Content is blocked based on the probability that it's harmful. The sensitivity level can be customized by choosing one of **Block few**, **Block some**, and **Block most** for each category. You can also get access to the **Block none** restricted option that disables RAI checks for the category after submitting a risk acknowledgment request (https://docs.google.com/forms/d/e/1FAIpQLSc9FEiXddsvrkiyqaQseZoZzCKu4iC7oHy1i0ZZ8uhrHLmJ1A/viewform) for your project and receiving approval.

For more information, see configure safety attributes (/vertex-ai/generative-ai/docs/multimodal/configure-safety-attributes).

  - **Prompt security** (#settings-generative-prompt-security)

    You can check the **enable prompt security check** setting to enable prompt security checks. When enabled, the agent will attempt to prevent prompt injection attacks. These attacks may be used to reveal parts of the agent prompt or to provide responses the agent is not supposed to supply. This is accomplished by sending an additional LLM prompt that checks whether the user query is possibly malicious.

  - Generative Fallback

    - **Text prompt configuration** (#settings-generative-fallback-prompt)

      See Generative fallback: define your own prompt (/dialogflow/cx/docs/concept/generative/generative-fallback#define_your_own_prompt).

- Data Store

  See Data store agent settings (/dialogflow/cx/docs/concept/data-store/settings).

## Deterministic Flows

These settings apply to all flows in this agent, except for the ML settings which are applied per flow. To edit other flow-specific settings, navigate to the flow in the console and edit the settings there.

- Intent training

Flows uses machine learning (ML) algorithms to understand end-user inputs, match them to intents, and extract structured data. Flows learn from training phrases (/dialogflow/cx/docs/concept/intent#tp) that you provide and the language models built into flows. Based on this data, it builds a model for making decisions about which intent should be matched to an end-user input. You can apply unique ML settings for each flow of an agent.

The following intent training settings are available:

- **Spell correction** (#spelling-correction)

  If this is enabled and end-user input has a spelling or grammar mistake, an intent will be matched as though it was written correctly. The detect intent response will contain the corrected end-user input. For example, if an end-user enters "I want an applle", it will be processed as though the end-user entered "I want an apple". This also applies to matches involving both system and custom entities.

  Spell correction is available in English, French, German, Spanish, and Italian. It is available in all Conversational Agents (Dialogflow CX) regions (/dialogflow/cx/docs/concept/region).

★ **Note:** If original and corrected end-user input match different intents, the intent matching the original end-user input is selected.

  Warnings and best practices:

  - Spell correction can't correct ASR (automatic speech recognition) errors, so we don't recommend enabling it for agents using ASR inputs.
  - It is possible for corrected input to match the wrong intent. You can fix this by adding commonly mismatched phrases to negative examples (/dialogflow/cx/docs/concept/intent#negative).
  - Spell correction increases the agent's response time slightly.
  - If an agent is defined using domain-specific jargon, the corrections may be undesired.

- Flow-specific ML settings

  - **NLU type** (#nlu-type)

    This can be one of:

    - **Advanced NLU** (default): Advanced NLU technology. This NLU type works better than standard, especially for large agents and flows.

        - **Standard NLU**: Standard NLU technology. Will no longer receive quality improvements or new features (/dialogflow/docs/release-notes#February_20_2024) .

    - **Auto train** (#auto-train)

      If enabled, the flow is trained whenever it is updated with the console. For large flows, this may cause console UI delays, so you should disable this setting and manually train (/dialogflow/cx/docs/concept/flow#train) as needed for large flows.

    - **Classification threshold** (#classification-threshold)

      To filter out false positive results and still get variety in matched natural language inputs for your agent, you can tune the machine learning classification threshold. This setting controls the minimum intent detection confidence required for an intent match (/dialogflow/cx/docs/concept/intent#match).

      If the confidence score for an intent match is less than the threshold value, then a no-match event (/dialogflow/cx/docs/concept/handler#event-built-in) will be invoked.

      You can set a separate classification threshold value for every flow in each language enabled for the agent. This is to accommodate different languages performing best at different classification thresholds. For more information about creating a multilingual agent, see the multilingual agents documentation (/dialogflow/cx/docs/concept/agent-multilingual).

      ★ **Note:** Before adjusting this value, you should try adding negative examples (/dialogflow/cx/docs/concept/intent#negative).

    - **Training status** (#training-status)

      Indicates whether the flow has been trained since the latest update to the flow data.

    - **Train** (#train-nlu)

      Use this button to manually train the flow.

- **Language** (#flow-language)

  The languages supported by your agent. Once an agent is created, the default language cannot be changed. However, you can perform the following:

    1. Export your agent to the JSON format (/dialogflow/cx/docs/reference/json-export).
    2. Extract the downloaded file.

3. Find the `agent.json` file.
4. Update the `defaultLanguageCode` and `supportedLanguageCodes` fields to the desired values.
5. Restore (/dialogflow/cx/docs/reference/json-export#restore) the agent to the same or different agent from step 1.
6. Update language-specific training phrases and entity values as needed.

You can also enable language auto detection and switching.

## Connectivity

- **Custom payload template** 🔗 (#custom-payload-template)

  In this section, you can create descriptions and payloads for custom payload templates (/dialogflow/cx/docs/concept/fulfillment#custom-payload-templates).

- **Parameter definition** 🔗 (#connectivity-parameter-definition)

  Clearly state what parameters from the session parameters will be sent to the webhook, and what session parameters can be set by the webhook's response.

- **Client certificate settings** 🔗 (#connectivity-client-certificate)

  Used to upload SSL certificates.

## Speech and IVR

- **Speech options**
  - **Text-to-Speech** 🔗 (#text-to-speech)
    - **Voice selection** 🔗 (#voice-selection)

      You can select the language and voice used for speech synthesis.

      You may enable Custom voice (/text-to-speech/custom-voice/docs) for your agent by selecting the custom voice option from the voice selection dropbox and specifying the custom voice name in the corresponding field. The custom voice name must follow the following pattern: `projects/PROJECT_ID/locations/LOCATION/models/MODEL_NAME`.

      You can also use voice cloning (/dialogflow/cx/docs/concept/voice-cloning).

      - If you are using telephony gateway, make sure the **Dialogflow Service Agent** service account `service-PROJECT_NUMBER@gcp-sa-dialogflow.iam.gserviceaccount.com` is granted with "AutoML Predictor" in your custom voice project.
      - For regular API calls, make sure the service account used to call Conversational Agents (Dialogflow CX) is granted with "AutoML Predictor" role in your custom voice project.
  - **Speech-to-Text** (#speech-to-text)
    - **Enable auto speech adaptation** (#auto-speech-adaptation)

      See Auto speech adaptation (/dialogflow/cx/docs/concept/speech-adaptation#auto-adaptation).
    - **Enable advanced speech settings** (#advanced-speech-settings)

      For details about advanced speech options, see the Advanced speech settings (/dialogflow/cx/docs/concept/advanced-speech) guide.
    - **DTMF settings** (#speech-dtmf)

      See DTMF for telephony integrations (/dialogflow/cx/docs/concept/dtmf) for more information.
    - **Enable manual speech adaptation** (#speech-manual)

      See manual speech adaptation (/dialogflow/cx/docs/concept/speech-adaptation#manual-adaptation).
- **Call companion**

  See Call companion (/dialogflow/cx/docs/concept/call-companion).

## UI Settings

You can use these settings to customize the Messenger and Call Companion user interfaces.

## Security

See Security settings (/dialogflow/cx/docs/concept/security-settings) and Access control (/dialogflow/cx/docs/concept/access-control).

## Dialogflow CX console agent settings

To access agent settings:

ConsoleAPI (#api)
(#console)

1. Open the console.
2. Choose your Google Cloud project.
3. Select your agent.
4. Click **Agent Settings**.
5. Update the settings as desired.
6. Click **Save**.

### General settings

The following general settings are available for agents:

- **Display name**

  A human-readable name for your agent.

- **Time zone**

  The default time zone for your agent.

- **Default language**

  The default language supported by your agent. Once an agent is created, the default language cannot be changed. However, you can perform the following:

  1. Export (#export) your agent to the JSON format (/dialogflow/cx/docs/reference/json-export).
  2. Unzip the downloaded file.
  3. Find the `agent.json` file.
  4. Update the `defaultLanguageCode` and `supportedLanguageCodes` fields to the desired values.
  5. Restore (/dialogflow/cx/docs/reference/json-export#restore) the agent to the same or different agent from step 1.

6. Update language-specific training phrases and entity values as needed.

- **Agent lock**
  - **Lock the agent**

    Indicates whether the agent is locked. A locked agent cannot be edited.
- **Logging settings**
  - **Enable Cloud Logging**

    Indicates whether Cloud logging (/dialogflow/cx/docs/concept/logging) is enabled for the agent.
  - **Enable interaction logging**

    Indicates whether you would like Google to collect and store redacted (/dialogflow/cx/docs/concept/security-settings) end-user queries for quality improvement.

★ **Note:** Some features, such as Experiments (/dialogflow/cx/docs/concept/experiments), Conversation history (/dialogflow/cx/docs/concept/conversation-history), BigQuery export (/dialogflow/cx/docs/concept/export-bq), Analytics (/dialogflow/cx/docs/concept/analytics), and NLU model improvement require this setting to be enabled, as they rely on traffic logs.

  - **Enable consent-based end-user input redaction**

    If this setting is enabled, it lets you use a special session parameter to control whether end-user input and parameters are redacted from conversation history (/dialogflow/cx/docs/concept/conversation-history) and Cloud logging (/dialogflow/cx/docs/concept/logging), by default the session parameter is `true`. If this setting is disabled, no redaction occurs.

    User consent is collected using a boolean session parameter: `$session.params.conversation-redaction`. If this setting is enabled, and the session parameter is set to `false`, no redaction occurs **(other redaction strategies still apply)**. If this setting is enabled, and the session parameter is set to `true`, redaction occurs.

    An example consent requesting flow could be: first ask the user if they would like to keep end-user input, and match the response with two intents, one is "yes intent" and the other is "no intent". Then, set the session parameter to `false` (no redaction) in the parameter presets (/dialogflow/cx/docs/concept/fulfillment#param-preset) of the "yes intent" route in fulfillment (/dialogflow/cx/docs/concept/fulfillment), and to `true` (redaction occurs) in the parameter preset (/dialogflow/cx/docs/concept/fulfillment#param-preset) of the "no intent" route.

- **BigQuery export**
  - **Enable BigQuery export**

    Indicates whether BigQuery export (/dialogflow/cx/docs/concept/export-bq) is enabled.
  - **BigQuery dataset**

    The BigQuery dataset name.
  - **BigQuery table**

    The BigQuery table name.
- **Intent Suggestions**

  You can enable intent suggestions (/dialogflow/cx/docs/concept/intent#suggestions).
- **Custom payload template**

  In this section, you can create descriptions and payloads for custom payload templates (/dialogflow/cx/docs/concept/fulfillment#custom-payload-templates).

## ML settings

Conversational Agents (Dialogflow CX) uses machine learning (ML) algorithms to understand end-user inputs, match them to intents, and extract structured data. Conversational Agents (Dialogflow CX) learns from training phrases (/dialogflow/cx/docs/concept/intent#tp) that you provide and the language models built into Conversational Agents (Dialogflow CX). Based on this data, it builds a model for making decisions about which intent should be matched to an end-user input. You can apply unique ML settings for each flow of an agent, and the model created by Conversational Agents (Dialogflow CX) is unique for each flow.

The following agent-wide ML settings are available:

- **Allow ML to correct spelling**

  If this is enabled and end-user input has a spelling or grammar mistake, an intent will be matched as though it was written correctly. The detect intent response will contain the corrected end-user input. For example, if an end-user enters "I want an applle", it will be processed as though the end-user entered "I want an apple". This also applies to matches involving both system and custom entities.

  Spell correction is available in English, French, German, Spanish, and Italian. It is available in all Conversational Agents (Dialogflow CX) regions (/dialogflow/cx/docs/concept/region).

★ **Note:** If original and corrected end-user input match different intents, the intent matching the original end-user input is selected.

Warnings and best practices:

- Spell correction can't correct ASR (automatic speech recognition) errors, so we don't recommend enabling it for agents using ASR inputs.
- It is possible for corrected input to match the wrong intent. You can fix this by adding commonly mismatched phrases to negative examples (/dialogflow/cx/docs/concept/intent#negative).
- Spell correction increases the agent's response time slightly.
- If an agent is defined using domain-specific jargon, the corrections may be undesired.

The following flow-specific ML settings are available:

- **NLU type**

  This can be one of:

  - **Advanced NLU** (default): Advanced NLU technology. This NLU type works better than standard, especially for large agents and flows.
  - **Standard NLU**: Standard NLU technology. Will no longer receive quality improvements or new features (/dialogflow/docs/release-notes#February_20_2024).

- **Auto train**

  If enabled, the flow is trained whenever it is updated with the console. For large flows, this may cause console UI delays, so you should disable this setting and manually train (/dialogflow/cx/docs/concept/flow#train) as needed for large flows.

- **Classification threshold**

  To filter out false positive results and still get variety in matched natural language inputs for your agent, you can tune the machine learning classification threshold. This setting controls the minimum intent detection confidence required for an intent match (/dialogflow/cx/docs/concept/intent#match).

  If the confidence score for an intent match is less than the threshold value, then a no-match event (/dialogflow/cx/docs/concept/handler#event-built-in) will be invoked.

  You can set a separate classification threshold value for every flow in each language enabled for the agent. This is to accommodate different languages performing best at different classification thresholds. For more information about creating a multilingual agent, see the multilingual agents documentation (/dialogflow/cx/docs/concept/agent-multilingual).

★ **Note:** Before adjusting this value, you should try adding negative examples (/dialogflow/cx/docs/concept/intent#negative).

- **Training status**

  Indicates whether the flow has been trained since the latest update to the flow data.

- **Train NLU**

  Use this button to manually train the flow.

## Generative AI settings

The following generative AI settings are available:

- General
  - **Banned phrases**

    List of phrases that are banned for generative AI. If a banned phrase appears in the prompt or the generated response, the generation will fail.

  - **Safety filters**

    Configure sensitivity levels of safety filters with respect to different Responsible AI (RAI) categories. Content will be assessed against the following four categories:

| Category | Description |
|---|---|
| Hate speech | Negative or harmful comments targeting identity and/or protected attributes. |
| Dangerous content | Promotes or enables access to harmful goods, services, and activities |
| Sexually explicit content | Contains references to sexual acts or other lewd content |
| Harassment | Malicious, intimidating, bullying, or abusive comments targeting another individual |

    Content is blocked based on the probability that it's harmful. The sensitivity level can be customized by choosing one of **Block few**, **Block some**, and **Block most** for each category. You can also get access to the **Block none** restricted option that disables RAI checks for the category after submitting a risk acknowledgment request (https://docs.google.com/forms/d/e/1FAIpQLSc9FEiXddsvrkiyqaQseZoZzCKu4iC7oHy1i0ZZ8uhrHLmJ1A/viewform) for your project and receiving approval.

For more information, see configure safety attributes (/vertex-ai/generative-ai/docs/multimodal/configure-safety-attributes).

- **Prompt security**

  You can check the **enable prompt security check** setting to enable prompt security checks. When enabled, the agent will attempt to prevent prompt injection attacks. These attacks may be used to reveal parts of the agent prompt or to provide responses the agent is not supposed to supply. This is accomplished by sending an additional LLM prompt that checks whether the user query is possibly malicious.

- Generative Agent

★ **Note:** This setting has restricted access.

- **Generative model selection**

  Select the model used by generative features. For more information, see model versions (/vertex-ai/generative-ai/docs/learn/model-versioning).

- **Playbook context truncation**

  **Playbook context truncation** culls some past turns from the playbook prompt in order to keep the prompt size from growing with every sequential turn handled by the playbook. This feature offers a way to mitigate unwanted prompt size growth.

  Normally, without truncation, each subsequent turn will be appended into the "conversation history" of the LLM prompt regardless of whether it is relevant to the current turn. This can ultimately lead to the prompt increasing in size with every turn. As more of the prompt is taken up by conversation history, less of the prompt can be used for few-shot examples (so these might get dropped). Eventually, the prompt might also breach current token limits. You can increase token sizes to accommodate this, but keep in mind that increased prompt sizes also add to the LLM response latency.

  Playbook context truncation lets you set a percentage of the token budget to be reserved for conversation history, as a maximum. Conversation turns are preserved in most recent to least recent order. This setting can help you prevent token limits from being exceeded. Regardless of which setting you choose, a minimum of two conversation turns are preserved, in most recent to least recent order.

You must first set a token limit (/dialogflow/cx/docs/concept/agent-settings#builder-settings-generative) before you can modify this setting.

**Important:** Truncating context might cause some parameters to be inadvertently be lost if they are part of culled turns. Evaluate your playbook interactions carefully after enabling this option.

Token input budget is also used by the following:

- System instructions and examples: Automatically added to the prompt. This behavior cannot be modified.
- Playbook instructions and goals: Any instructions and goals that you write will be added to the prompt in their entirety.
- Playbook few-shot examples: Are added either in order (by default) or by an algorithm that you choose (such as regular expression best match ordering). Examples are culled to fit within input token budget after all other items are included.
- Conversation history made up of user and agent utterances, flow and playbook transition context, tool calls and outputs in the same session from all previous turns sequentially handled by the current playbook.

- Generative Fallback
  - **Text prompt configuration**

    See Generative fallback: define your own prompt (/dialogflow/cx/docs/concept/generative/generative-fallback#define_your_own_prompt).
- Data Store

  See Data store agent settings (/dialogflow/vertex/docs/concept/data-store-agent-settings).

## Speech and IVR settings

The following speech and IVR settings are available:

- **Text-to-Speech**
  - **Voice selection**

    You can select the language and voice used for speech synthesis.

You may enable Custom voice (/text-to-speech/custom-voice/docs) for your agent by selecting the custom voice option from the voice selection dropbox and specifying the custom voice name in the corresponding field. The custom voice name must follow the following pattern: `projects/PROJECT_ID/locations/LOCATION/models/MODEL_NAME`.

  - If you are using telephony gateway, make sure the **Dialogflow Service Agent** service account `service-PROJECT_NUMBER@gcp-sa-dialogflow.iam.gserviceaccount.com` is granted with "AutoML Predictor" in your custom voice project.
  - For regular API calls, make sure the service account used to call Conversational Agents (Dialogflow CX) is granted with "AutoML Predictor" role in your custom voice project.

- **Speech-to-Text**
  - **Enable auto speech adaptation**

    See Auto speech adaptation (/dialogflow/cx/docs/concept/speech-adaptation#auto-adaptation).
  - **Advanced speech settings**

    For details about advanced speech options, see the Advanced speech settings (/dialogflow/cx/docs/concept/advanced-speech) guide.

## DTMF

See DTMF for telephony integrations (/dialogflow/cx/docs/concept/dtmf) for more information.

## Multimodal

See Call companion (/dialogflow/cx/docs/concept/call-companion).

## Share settings

See Access control (/dialogflow/cx/docs/concept/access-control).

## Languages settings

Add additional language support to your agent. For the full list of languages, see the language reference (/dialogflow/cx/docs/reference/language).

#### Language auto detection

When you configure language auto detection (/dialogflow/cx/docs/concept/agent-multilingual#lang-detect), your chat agent will automatically detect the end-user's language and switch to that language. See the language auto detection documentation (/dialogflow/cx/docs/concept/agent-multilingual#lang-detect) for details.

### Security settings

See Security settings (/dialogflow/cx/docs/concept/security-settings).

### Advanced settings

Currently, the only advanced setting is for sentiment analysis (/dialogflow/cx/docs/concept/sentiment).

## AI Applications console settings

This section describes the settings available for agent apps.

### General

The following general settings are available for agent apps:

- **Display name**

  A human-readable name for your agent app.

- **Location**

  The agent app region.

- **App lock**

  If enabled, changes to the agent app are not permitted.

### Logging

The following logging settings are available for agent apps:

- **Enable Cloud Logging**

  If enabled, logs will be sent to Cloud Logging (/logging/docs).

- **Enable Conversation History**

  If enabled, conversation history (/dialogflow/vertex/docs/concept/user-interface#conversation-history) will be available. Indicates whether you would like Google to collect and store redacted end-user queries for quality improvement. This setting does not affect whether conversation history is used to generate agent responses.

- **Enable BigQuery Export**

  If enabled, conversation history is exported to BigQuery. The **Enable Conversation History** setting must also be enabled.

## GenAI

The following generative AI settings are available for agent apps:

- **Generative model selection**

  Select the generative model (/vertex-ai/generative-ai/docs/learn/model-versioning) that agents should use by default.

- **Input token limit**

  Select the input token limit for the generative model (/vertex-ai/generative-ai/docs/learn/models). This is the maximum token size for input sent to the model. Depending on the model, a token (/vertex-ai/generative-ai/docs/multimodal/get-token-count) can be somewhere between one character and one word. Smaller token limits have lower latency, but the model input size is limited. Larger token limits have higher latency, but the model input size can be larger.

- **Output token limit**

  Select the output token limit for the generative model (/vertex-ai/generative-ai/docs/learn/models). This is the maximum token size for output received from the model. Depending on the model, a token (/vertex-ai/generative-ai/docs/multimodal/get-token-count) can be somewhere between one character and one word. Smaller token limits have lower latency, but the model output size is limited. Larger token limits have higher latency, but the model output size can be larger.

- **Temperature**

  The temperature for a LLM lets you control how creative the responses are. A low value provides more predictable responses. A high value provides more creative or random responses.

- **Banned phrases**

List of phrases that are banned for generative AI. If a banned phrase appears in the prompt or the generated response, the agent will return a fallback response instead.

- **Safety filters**

  Configure sensitivity levels of safety filters with respect to different Responsible AI (RAI) categories. Content will be assessed against the following four categories:

  | Category | Description |
  |---|---|
  | Hate speech | Negative or harmful comments targeting identity and/or protected attributes. |
  | Dangerous content | Promotes or enables access to harmful goods, services, and activities |
  | Sexually explicit content | Contains references to sexual acts or other lewd content |
  | Harassment | Malicious, intimidating, bullying, or abusive comments targeting another individual |

  Content is blocked based on the probability that it's harmful. The sensitivity level can be customized by choosing one of **Block few** (blocking only high-probability instances of harmful content), **Block some** (medium and high probability instances), and **Block most** (low, medium, and high probability) for each category. You can also get access to the **Block none** restricted option that disables RAI checks for the category after submitting a risk acknowledgment request (https://docs.google.com/forms/d/e/1FAIpQLSc9FEiXddsvrkiyqaQseZoZzCKu4iC7oHy1i0ZZ8uhrHLmJ1A/viewform) for your project and receiving approval.

  For more information, see configure safety attributes (/vertex-ai/generative-ai/docs/multimodal/configure-safety-attributes).

- **Prompt security**

  You can check the **enable prompt security check** setting to enable prompt security checks. When enabled, the agent will attempt to prevent prompt injection attacks. These attacks may be used to reveal parts of the agent prompt or to provide responses the agent is not supposed to supply. This is accomplished by sending an additional LLM prompt that checks whether the user query is possibly malicious.

## Git

These settings provide a Git integration. Follow the instructions to configure the integration.

Previous
← Zip code collection (/dialogflow/cx/docs/concept/prebuilt-component/zip-code-collection)

Next
Speech models (/dialogflow/cx/docs/concept/speech-models) →

Last updated 2025-06-05 UTC.