

User Experience Design Guide for Webex Assistant Skill Developers

Updated September 2020

What is Webex Assistant?

Webex Assistant for Webex Rooms lets people use their voice to complete a variety of tasks. Users can start a meeting, join a Personal Room, call anyone in their company directory, and more from any meeting space with a compatible device.

To learn more about what Webex Assistant for Rooms can do, ask “OK Webex, what can you do?” or visit cs.co/GetStarted-WXAssistant.

Webex Assistant Skills

With Webex Assistant Skills, you can add functionality to Webex Assistant for Rooms by creating custom skills. For scenarios where your users have their hands full, prefer not to touch a shared device, or could save time with voice commands, a custom skill may be a great solution. In this guide, we’ll walk you through some important principles and best practices to help you create user-friendly skills.

Using This Guide

To move between pages, use the arrows on your keyboard. To resize the guide, choose a setting from the Options menu at the top right.

1.

How Users Interact with Skills

Starting a Conversation

- Starting a Conversation with Webex Assistant
- Starting a Conversation with Webex Assistant Skills
- Open a Skill
- Ask a Skill

Continuing a Conversation

Ending a Conversation

- Exit
- Dismiss

2.

Naming Your Skill

Branding

Speech Recognition

Length and Frequency

3.

Conversation Design for Your Skill

Structuring Conversations

- Intro
- Talking with the User
- In-Skill Help
- Hints
- Outro

Writing Natural Language Responses

- NLR Length
- Conversational Flow
- Tone
- Word Choice

Error Handling

Topics

1.

How Users Interact with Skills

As the developer, Webex Assistant Skills empower you to provide your users with even more voice functionality. In this section, we'll cover the high-level ways users will interact with your skill: starting a conversation, continuing a conversation, and ending a conversation. To learn more about developing the conversations users will have with your skill, check out [Conversation Design for Your Skill](#).

Starting a Conversation

It's easy for users to switch between using Webex Assistant and a Webex Assistant Skill. To use a skill, they'll just need to include the skill's name in their command. Let's take a closer at the differences:

Starting a Conversation with Webex Assistant

Each time a user starts a conversation with Webex Assistant or a Webex Assistant Skill, they'll need to begin with a wake word. The wake word is a phrase that wakes the assistant from sleep mode, so it's ready to help. Webex Assistant's wake words are "OK Webex" and "Hey Webex," and aren't configurable. Following the wake word, ask the assistant a question or tell the assistant what you need—that's called the query. For example:

 *User: "OK Webex, what can you do?"*

Starting a Conversation with Webex Assistant Skills

To use a Webex Assistant Skill, add a skill invocation phrase when beginning a conversation. The skill invocation phrase will always follow the wake word, and includes an invoke verb and the skill name. The invoke verbs aren't configurable, but you'll define your skill's name (take a look at [Naming Your Skill](#) for more details). Conversations with your skill will always begin with the wake word and an invocation phrase:

🗨️ *User: “OK Webex, <invoke verb> <SkillName>”*

Webex Assistant Skills gives users two ways to interact with your skill: ‘open a skill’ or ‘ask a skill.’ When a user is new to using your skill, opening it is a great way to explore how it works and learn about the skill’s functionality. When first using your skill, users might not know what it can do. We’ve found that offering a way for users to be introduced and to play around with the skill helps their understanding, empowering them to become an advanced skill user. Once users are more familiar with your skill, they can use ‘ask a skill’ to ask a question or tell it what they'd like it to do to. This works well for more advanced users who no longer need an introduction to your skill.

Open a Skill

To open the skill, use either “open” or “talk to” as the invoke verb. Following the same pattern above, a user would use a wake word, an open invoke verb, and the skill name.

Open invoke verbs: “open,” “talk to”

🗨️ *User: “OK Webex, open <SkillName>”*

🗨️ *User: "OK Webex, talk to <SkillName>"*

Ask a Skill

To ask the skill a question, use “ask” or “tell” as the invoke verb. When users are asking a skill something, their query is included, too.

Open invoke verbs: “ask,” “tell”

🗨️ *User: “OK Webex, ask <SkillName> <query>”*

🗨️ *User: “OK Webex, tell <SkillName> <query>”*

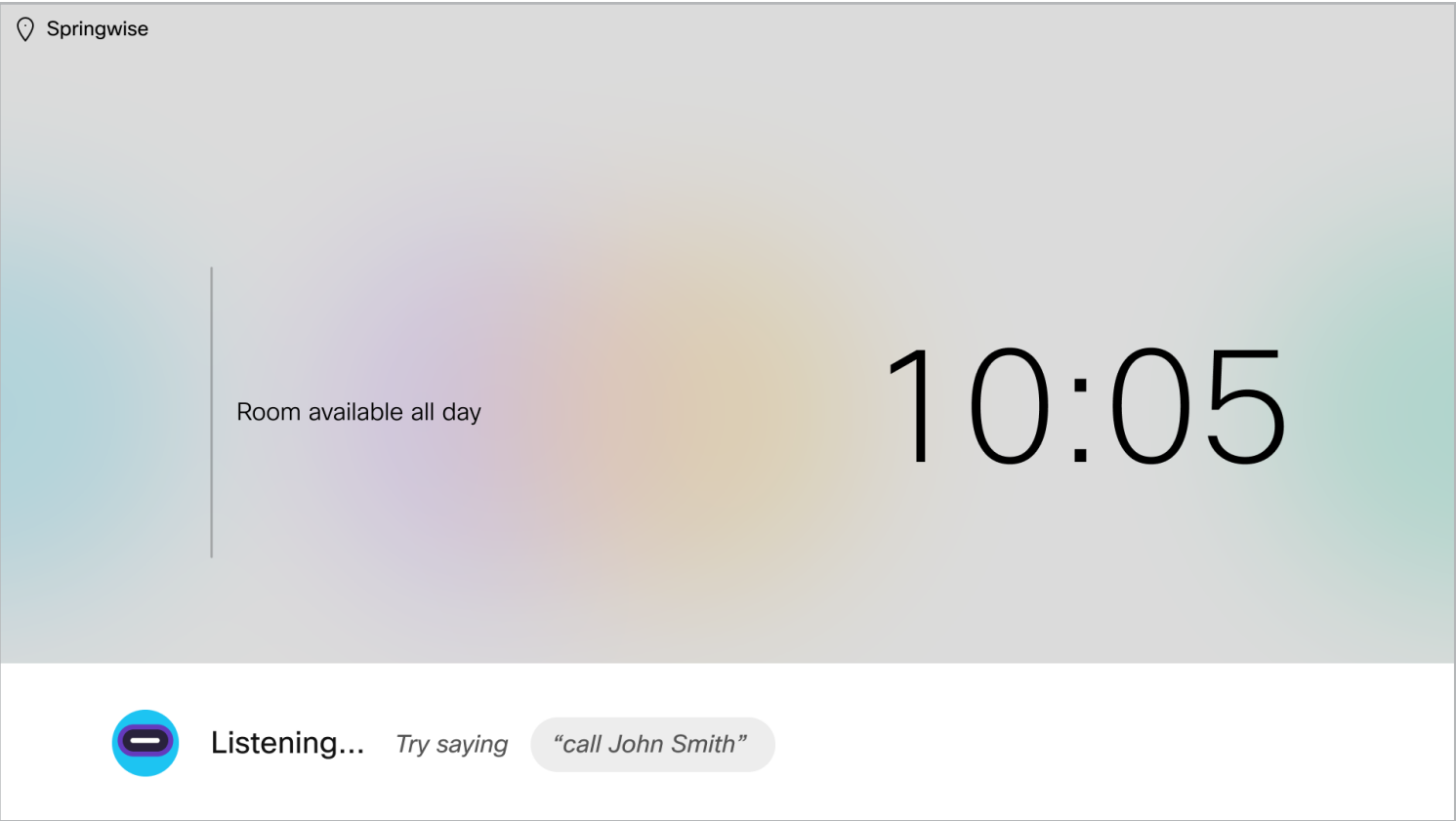
Remember, all conversations with your skill will begin with a wake word and a skill invocation phrase.

Continuing a Conversation

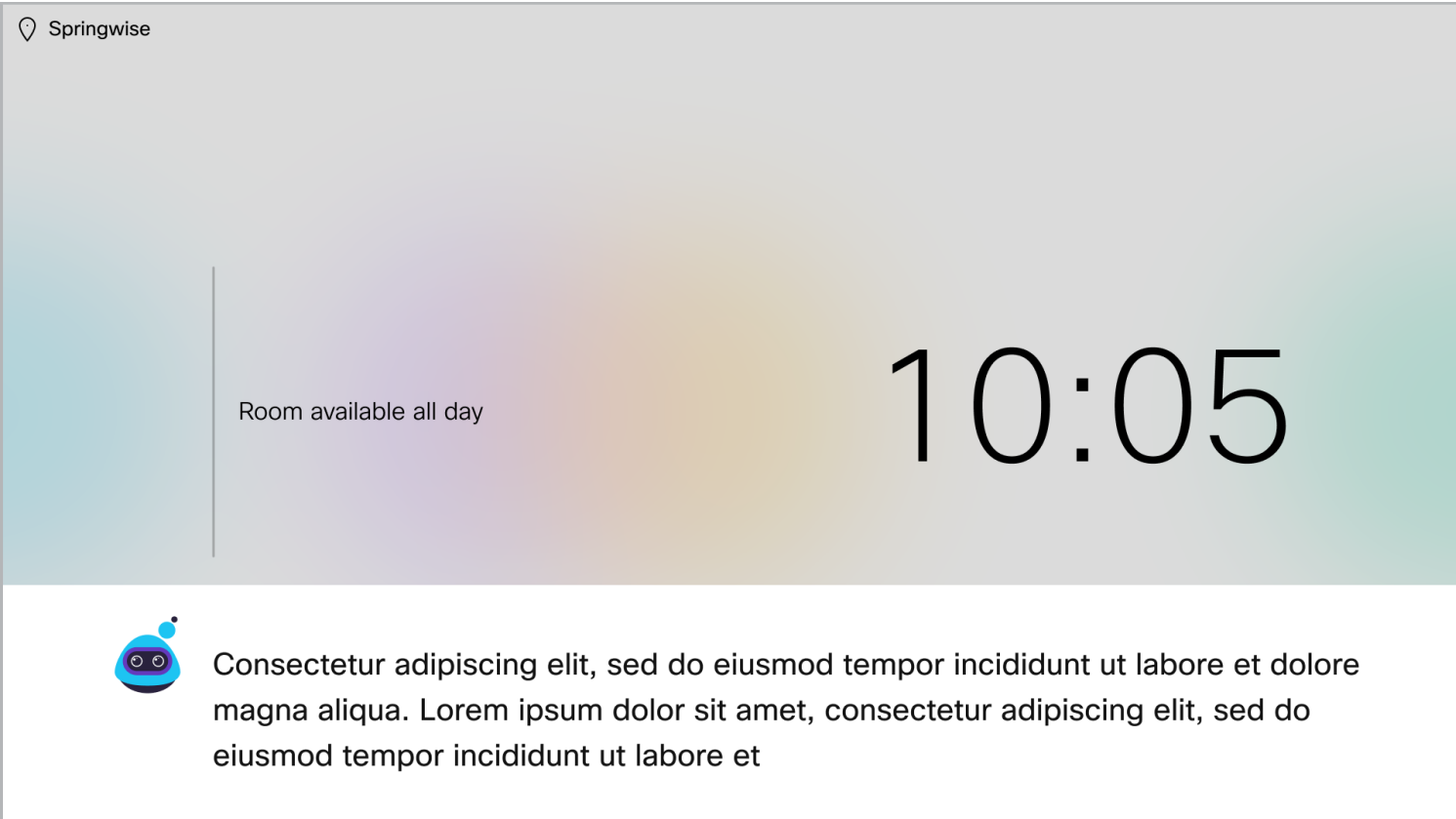
After starting a conversation with your Webex Assistant Skill, the assistant can be in ‘listening’ or ‘not listening’ mode. When the assistant is listening, the user can keep speaking without having to use a wake word or an invocation phrase to talk with your skill. When the assistant isn't listening, the user will need to use a wake word and invocation phrase to continue using your skill.

The Webex Assistant icon indicates if the assistant is listening.

When the assistant is listening, you'll see the icon below. When you see this icon, you can keep talking:



When the assistant isn't listening, you'll see the icon below. When you see this icon, use a wake word and invocation phrase to keep using your Webex Assistant Skill:



Let's look at some conversations with our example skill, the Weather skill, to better understand what interacting with 'listening' and 'not listening' look like.

In this conversation, the Weather skill keeps listening:

🗣️ *User: "OK Webex, ask <SkillName> <query>"*



🤖 *Weather skill: "Where should I check the weather?"*

🗣️ *User: "San Francisco"*

🤖 *Weather skill: "It's 68° and partly cloudy in San Francisco today."*

When the skill asks a question, it's in listening mode because it's waiting for the user to provide more information. After the skill finishes the user's request—in this case, telling them the weather in San Francisco—it stops listening. The user's request has been fulfilled, and the conversation ends. If the user wanted to ask the Weather skill for the forecast somewhere else, they would need to start a new conversation using a wake word and skill invocation phrase.

In the conversation below, the skill doesn't keep listening. The skill has completed the user's request, and the conversation ends:

 *User: “OK Webex, ask Weather to check the weather in San Francisco for me”*
 *Weather skill: “It’s 68° and partly cloudy in San Francisco today.”*

Defining whether your skill will be in listening or waiting mode after your skill processes a query is something you control. To learn more about creating different conversation flows, check out [Conversation Design for Your Skill](#).

Ending a Conversation

A user's conversations with your skill might come to a natural end, the user might want to end the interaction, or the user may no longer need the assistant to be present. For the latter two cases, there are keywords that allow users to end a conversation with the skill.

There should be two keywords that allow users to stop interacting with your skill: “exit” and “dismiss.” Both keywords will end a conversation, but they'll have slightly different behaviors. As the developer, you must create the "exit" keyword/intent. The “dismiss” keyword is global, and your skill will inherit it from Webex Assistant.

Exit

We require skills developers to create an “exit” intent. The “exit” keyword should end a conversation and display the outro message, if you've configured one. We also require that the “exit” keyword close anything that you choose to display on-screen. “Exit” is the preferred way for users to end interactions with your skill because it shows your outro message, which is a helpful way to remind users about how they can use your skill again.

Here’s an example where the “exit” keyword is used to close the assistant:

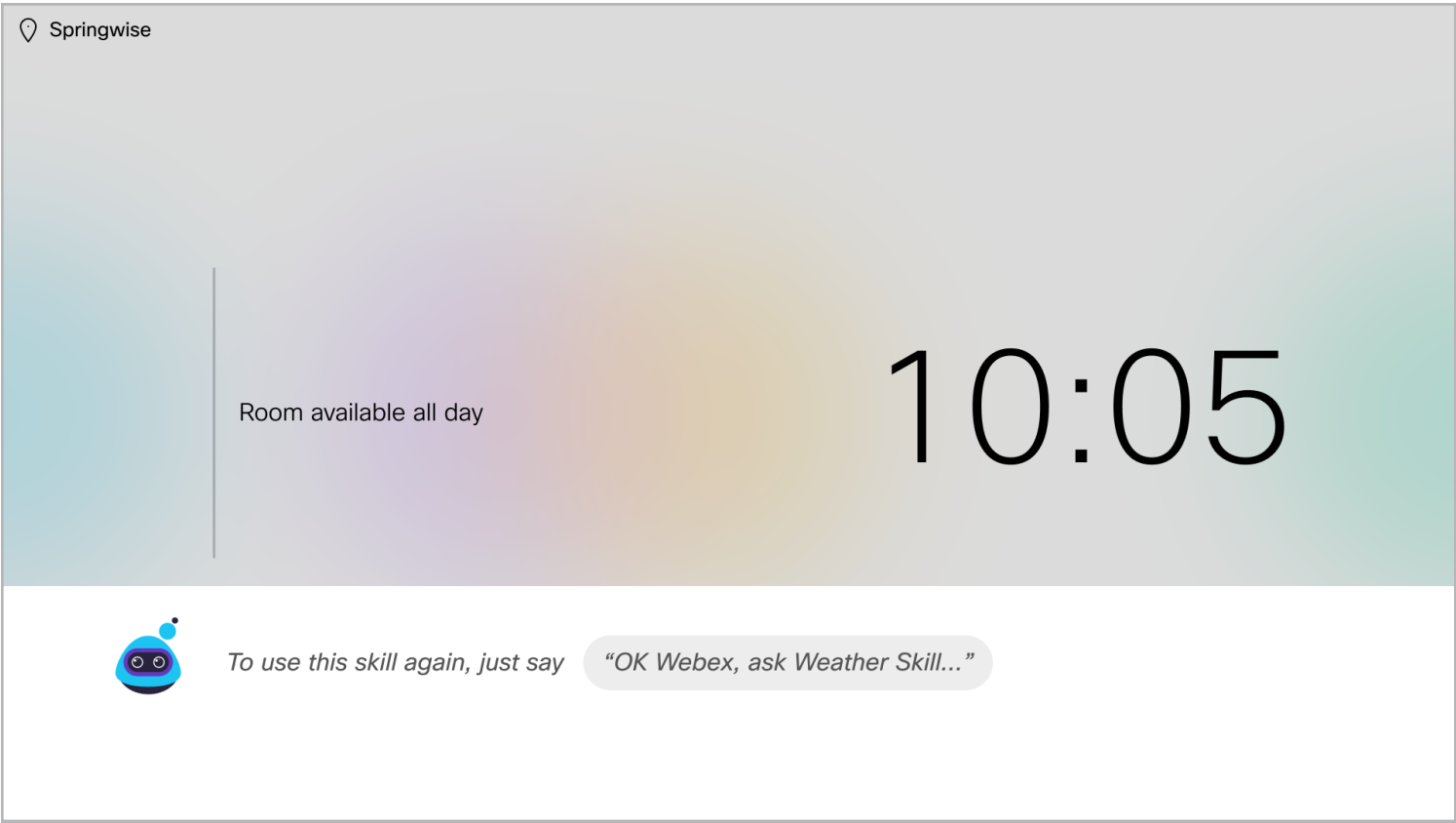
- 🗣️ *User: “OK Webex, ask Weather what the forecast is for this week”*
- 🤖 *Weather skill: “OK, here’s this week’s forecast: On Monday, the highs will be in the 80s and drop to the 50s in the evening. On Tuesday...”*
- 🗣️ *User, interrupting the skill: “OK Webex, tell Weather to exit”*
- 🤖 *Weather skill:*

Dismiss

The “dismiss” keyword will immediately end the conversation, and the outro message will not be displayed. You can think of “dismiss” as a stronger version of “exit.” A user might use “dismiss” if they need to end an interaction quickly or something is going wrong. This keyword allows the user to end a non-optimal interaction with ease. mple where the “exit” keyword is used to close the assistant:

- 🗣️ *User: “OK Webex, ask Weather for the forecast in Eventide”*
- 🤖 *Weather skill: “Hmmm, where should I check the forecast?”*
- 🗣️ *User: “Actually, dismiss”*

When users use the “exit” and “dismiss” keywords to end interactions, they'll need to make sure the assistant is listening. As with any other type of query, if the assistant isn't already listening, they'll need to use the wake word and skill invocation phrase.



2.

Naming Your Skill

Your skill's name needs to work consistently in the system while being memorable and intuitive for your users.

Branding

Choose a skill name that your users will associate with your skill's functionality and your workplace's positive traits. For example, a skill that provides information about your facilities probably shouldn't be named "Gladys" or "Company Bot." Neither of these indicate the service provided, and, for some people, "Gladys" could be confused with the malevolent AI GLaDOS, while "Bot" is associated with unhelpful automation. Names like "Help Desk" or "Office Manager" clearly communicate the skill's role and scope.

Depending on your design preference, the name can make the skill sound more lifelike (e.g., "Bartleby," "Electra"), moderately lifelike (e.g. "Office Manager," "Meeting Roomba") or more institutional (e.g., "Weather," "Human Resources").

Speech Recognition

The speech recognition engine should be able to consistently transcribe your skill name. Common alternate spellings can be added as skill name synonyms, but might not be desired for branding reasons.

Brand names that aren't found in the dictionary generally won't work well. If you name your skill "Deskly" or "Weatherator," the speech recognition won't write it that way and will likely alternate between different spellings.

It's helpful to try out the skill names you're considering in the context of the full invocation phrase. For example, if the skill name is “Vision," when a user says “OK Webex, tell Vision to start the coffeemaker," the speech recognition may transcribe it as “television." If the skill name is “Orbit," the speech recognition may transcribe “ask Orbit" as “ascorbic."

Try getting a few people with different accents to speak the invocation phrase to Webex Assistant to verify that the invocation phrase is usually written correctly. Once your skill name is registered, it will be added to the context phrases for the speech recognition within the organization, which will improve the accuracy. Still, a good skill name should typically work even without that.

If you plan to make the skill support multiple languages, you should also consider if the skill name will be recognized in each of the supported languages. Separate skill names for each language are acceptable, but may be undesirable.

Length and Frequency

Finally, the skill name shouldn't be too long, too common, or too rare.

For skill names that are common and short, there is some risk that a user would invoke it by accident. For example, if the skill is named “B" or “Tom," a user could say “talk to Bea" or “tell Tom to create a ticket," intending to use a core Webex Assistant feature, and your skill would appear instead.

Skill names that are long and uncommon will result in a lot of alternate spellings. Users may find it annoying to say the whole name each time, and a long name will crowd the on-screen hints and replies. To limit complexity, skill names and synonyms should be less than three words long as spelled by the speech recognition. For visual simplicity, we recommend that skill names be less than 25 characters.

3.

Conversation

Design for Your Skill

When users can have seamless conversations with your skill, it's better able to help them achieve their goals. How your conversations are structured, how the assistant sounds, and how you help users recover are all crucial parts of your skill.

First, we recommend thinking through structuring conversations—what conversational pieces are there, how you should build them, and how to get information from users. Next, we'll dive into the details of how to write Natural Language Responses (NLRs). How you word your replies will depend on where they occur in the larger conversational structure. Finally, we'll discuss error messages and how to word them to be most helpful to your users.

Structuring Conversations


Conversations with your skill, like any conversation you'd have with a friend or colleague, follow a conversational structure. Let's look at the pieces of a conversation that will allow users to have great conversations with your skill:

- Intro
- Talking with the User
- In-Skill Help
- Hints
- Outro

Usually there will be a start to the conversation, some back and forth, and a goodbye. We'll refer to these as the 'intro,' 'talking with the user,' and the 'outro.' We also recommend including in-skill help, hints, and error handling. In-skill help instructs users about what your skill can do and hints are on-screen help to guide users. These typically won't all appear in one conversation, but each piece is important to create a robust skill.


Intro

When a user opens your skill, it's helpful to offer a greeting and let the user know what the skill can do. We recommend including the skill name and a high-level example in your intro. To keep things conversational, we also recommend that your skill be in listening mode after speaking the intro, and that the intro ends with a question to let users know you're ready for their input:

 *Weather skill: “Welcome to the <skillname> skill! I can help you <action> and <action>. What can I do for you?”*

Our example skill, the Weather skill, can tell you today's forecast, tomorrow's forecast, and the temperature. Here's what an introduction for that skill might sound like:

 *User: “OK Webex, open Weather”*

 *Weather skill: “Welcome to the Weather skill! I can help you check the forecast and temperature. What can I help you with?”*

Talking with the User

Your skill will interact with your users by replying or having a back-and-forth conversation. In this section, we'll make recommendations for structuring conversations to communicate effectively and get the information you need to be able to help.

Users start having a conversation with your skill after they invoke it with the skill invocation phrase. After your skill is invoked, it's time to start conversing with your user.

When you're thinking about having a conversation with your user, you'll first need to think about what you want to accomplish with your skill. What will it help users do? The tasks that you'll help your users with are your skill's use cases. First, make a list of your use cases.

For example, if we're developing a skill that helps users get weather information, you might have the following use cases:

- Check the weather: get the weather for the user's current location
- Check the weather somewhere else: get the weather for another location

Next, list the information you need from the user to fulfill these requests. To check the weather, we need to know where the user is. For now, let's assume that we already know where the user is.

When you don't need information from the user, your skill can fulfill the user's request in a single turn:

💡 *User: “OK Webex, ask Weather to get me today’s forecast”*

🤖 *Weather skill: “OK! Today’s forecast for San Francisco is a high of 75° and a low of 56°”*

What about when you do need information from the user? In our example, the "check weather somewhere else" feature requires that the skill know where "somewhere else" is. When you need information from the user, we recommend that your skill ask the user for the required information. This is a great way to make your skill feel more conversational, and it helps users learn what your skill needs to be able to complete a request.

💡 *User: “OK Webex, ask Weather to show the weather for somewhere else”*

🤖 *Weather skill: “Where would you like me to check the weather?”*

💡 *User: “New York”*

🤖 *Weather skill: “OK! Today’s forecast for New York is a high of 52° and a low of 33°”*

We recommend using questions and multiple turns between the user and your skill to make the experience engaging. This also helps avoid scenarios where the user doesn't know what to provide, and your skill fails because it doesn't have the necessary information. For example, if checking the weather somewhere else didn't ask a follow-up question, your interaction might look like this:

💡 *User: “OK Webex, check the weather somewhere else”*

🤖 *Weather skill: “Hmmm, I can’t check the weather somewhere else unless I know where to check.”*

In this interaction, there's no dialogue between the user and the skill, and it forces the user to re-start the conversation.


When you need information from your user to complete their request, we recommend asking a question or a series of questions to get the information.

In-Skill Help

In-skill help is an important part of the conversation flow—it's how users will learn what your skill offers. We recommend creating an intent to capture different types of in-skill help queries like “how can you help me?” and “what can you do?”

We recommend that your in-skill help reply be a one-sentence summary that highlights your skill's key functionalities. There's no built-in UI for in-skill help, but if you'd like to show a web view or create something, please do. Even if you choose to show something on-screen, it's helpful if it's accompanied by a help reply that includes the skill name and 1-3 short examples.

The Weather skill can tell you the forecast and current temperature. In-skill help for the Weather skill might sound like:

 *Weather skill: "The Weather skill can help you check today's forecast and give you the current temperature."*

Note that in-skill help is different from the Webex Assistant Skills help page. In-skill help is specific to your skill and is where you showcase your skill's features. The Webex Assistant Skills help page only shows a list of the available skills. In the future, we hope to provide a way for you to add your skill to our help view. For now, use in-skill help to show off what your skill can do.

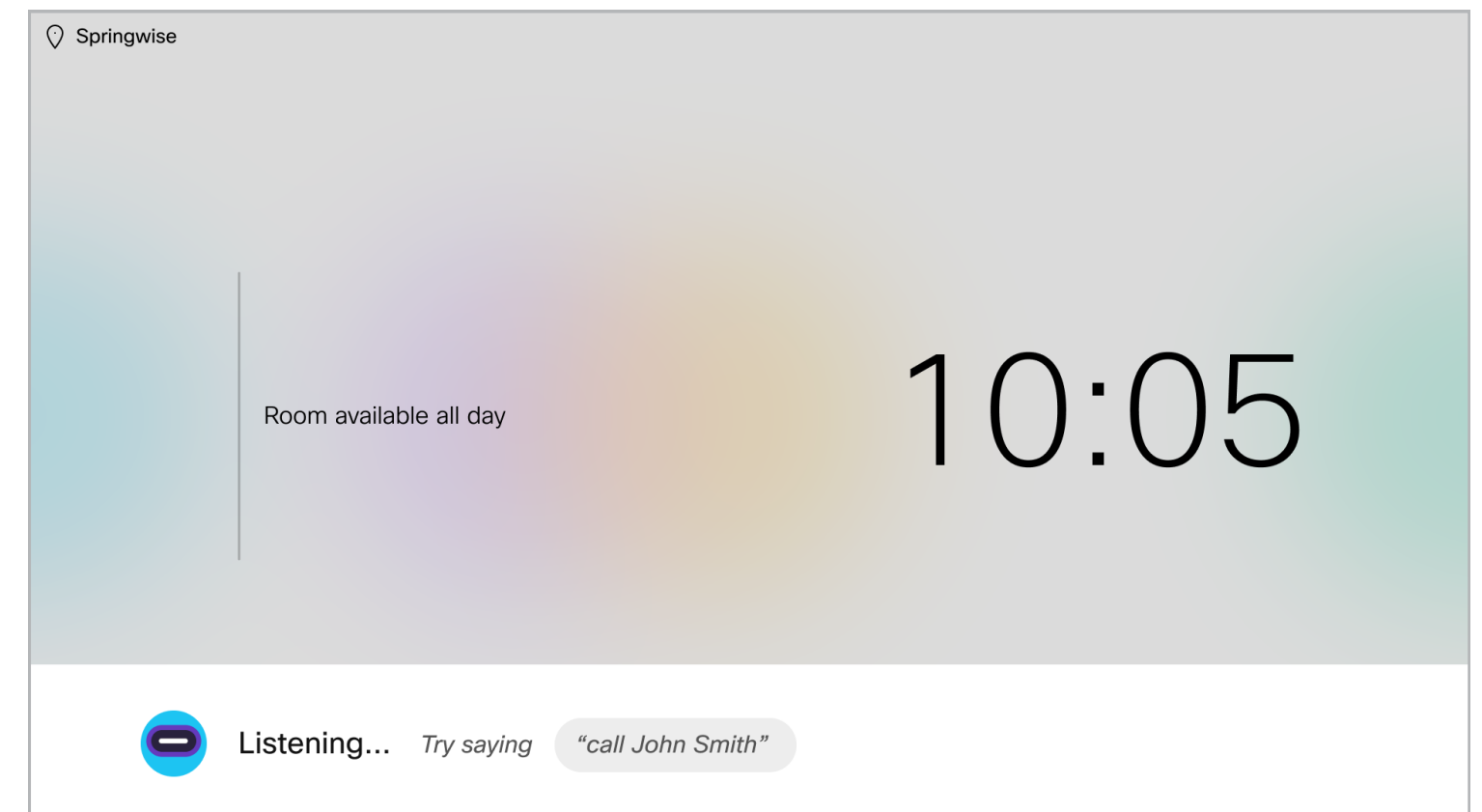
Hints

Because users typically don't fully understand an AI's capabilities or the right way to frame a query, they often aren't clear on how they can talk to it. Showing hints while your users are talking to your skill can help solve this problem.

1. What Is a Hint

A hint is a short piece of text displayed on the UI that suggests voice commands to the user. Hints can appear when the assistant is listening or not listening. Unlike replies, hints are never spoken by the assistant. They're an on-screen example to help nudge users in the right direction.

Webex Assistant usually shows hints when the assistant is listening. After the user says "OK Webex," the assistant wakes up and, while listening, shows a hint of an action the user might take. In the example below, since we don't have any context, we show a generic hint: "call John Smith."



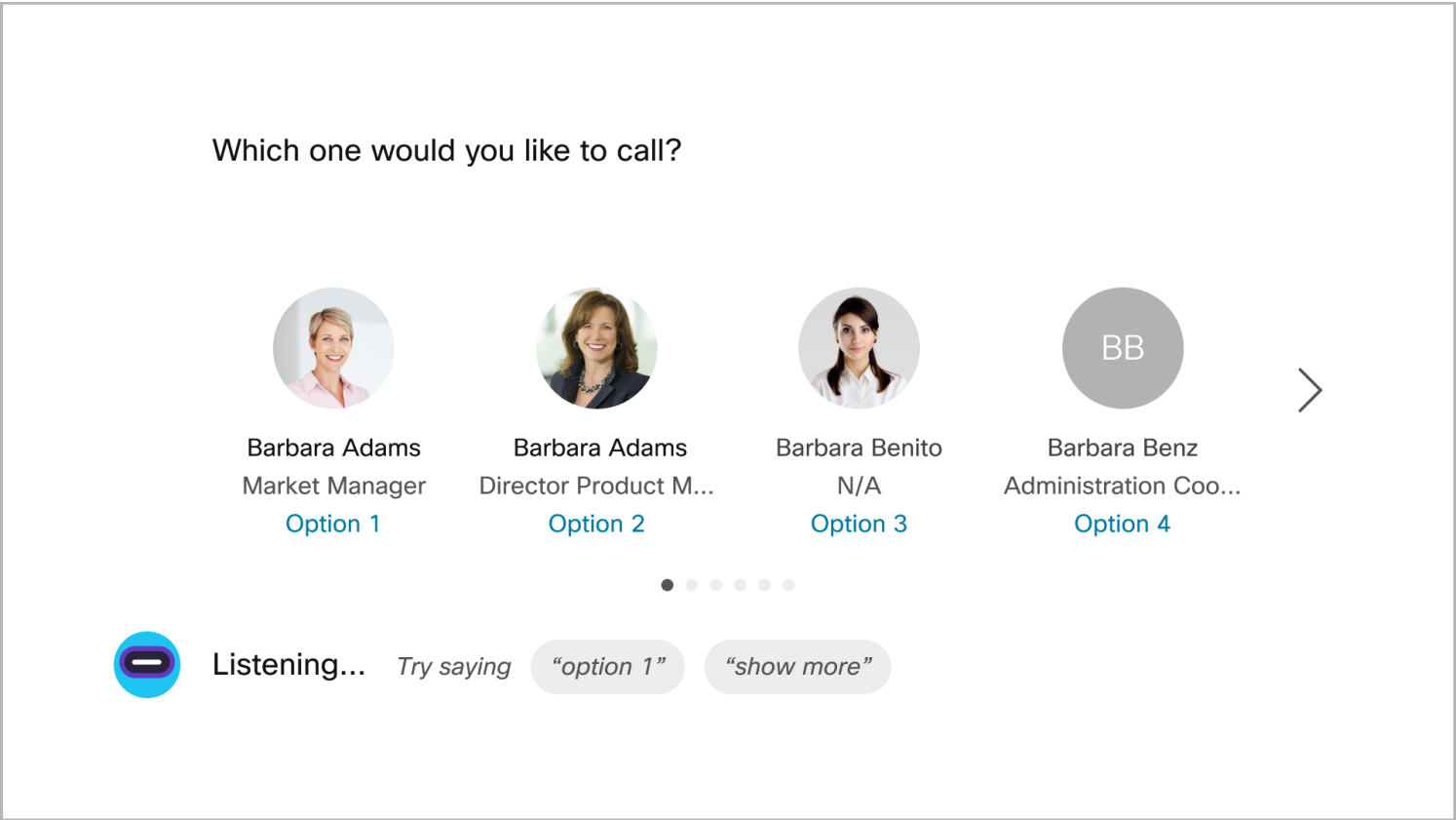
We recommend showing a generic hint when your skill is in listening mode. For example, the Weather skill might show a hint like this:

 Try saying “Show today’s forecast”

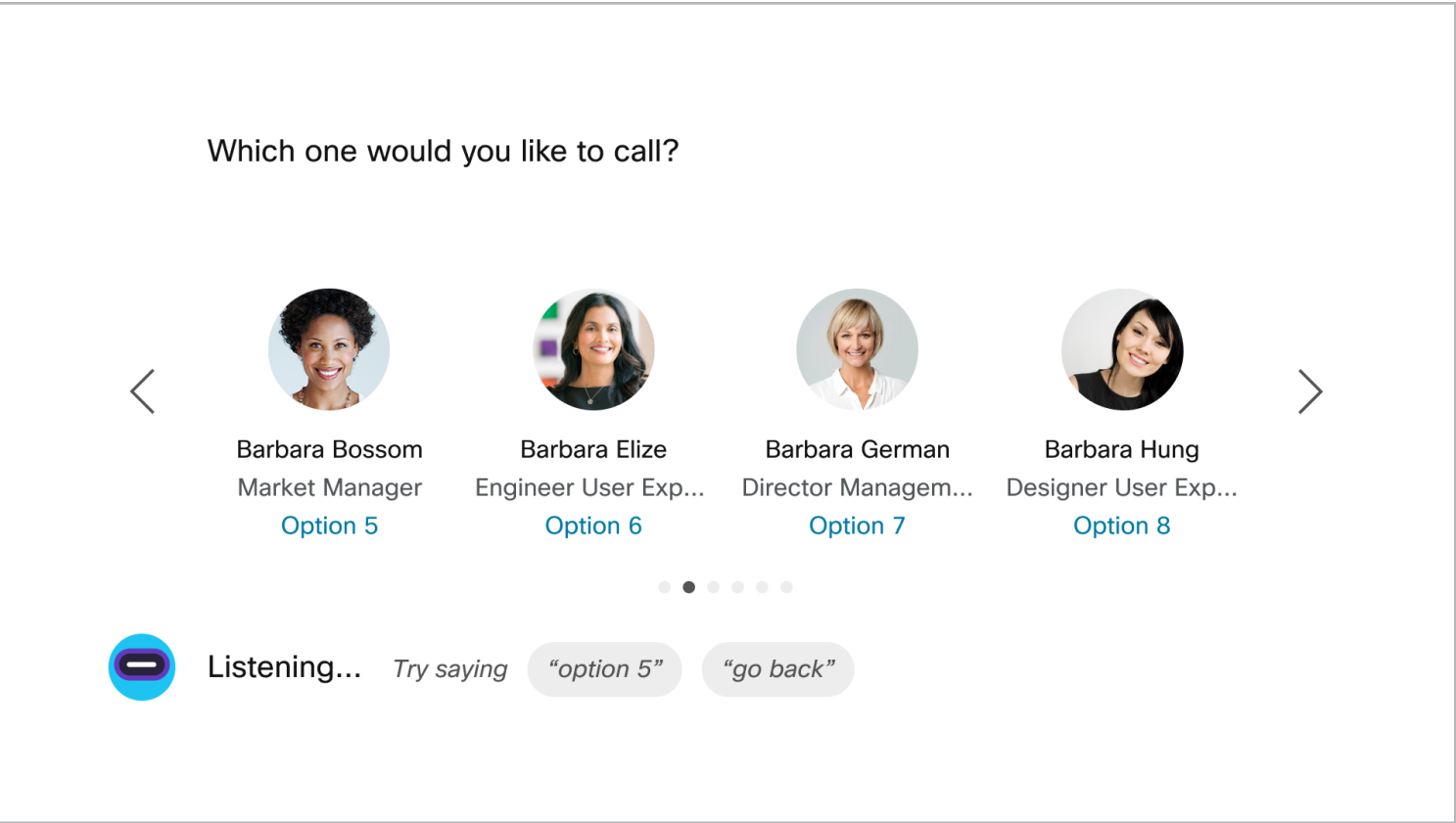
By displaying a hint, we can promote the assistant’s capabilities and offer something for users to play with. We've learned that when it comes to voice products, people learn best by playing around with them.

2. How to Use Hints

The most helpful hints indicate the main ways your user can interact with your skill and are relevant in the current context. For example, when users use Webex Assistant's 'call by name' feature, the hints displayed change depending on the context.



When a user lands on the first page of the option carousel, we show “option 1” and “show more.” This helps users understand how they can make a selection or scroll through the carousel to see more options.



The hints that you show when the assistant isn't listening are very important. If the skill is going to dismiss itself after giving a response, we suggest adding **To use this skill again, just say “OK Webex, ask <your skill name>...”** so that your users understand that they’ll have to include the skill's name in their query to interact with your skill again. Educating users about how and when to use the skill invocation phrase is crucial.

While hints are great for educating your users about what your skill can do, you should avoid using too many hints at once. This is visually overwhelming and will likely result in users ignoring the hints. We suggest only showing one or two hints at a time to provide suggestions without overwhelming your users.

3. How to Phrase a Hint

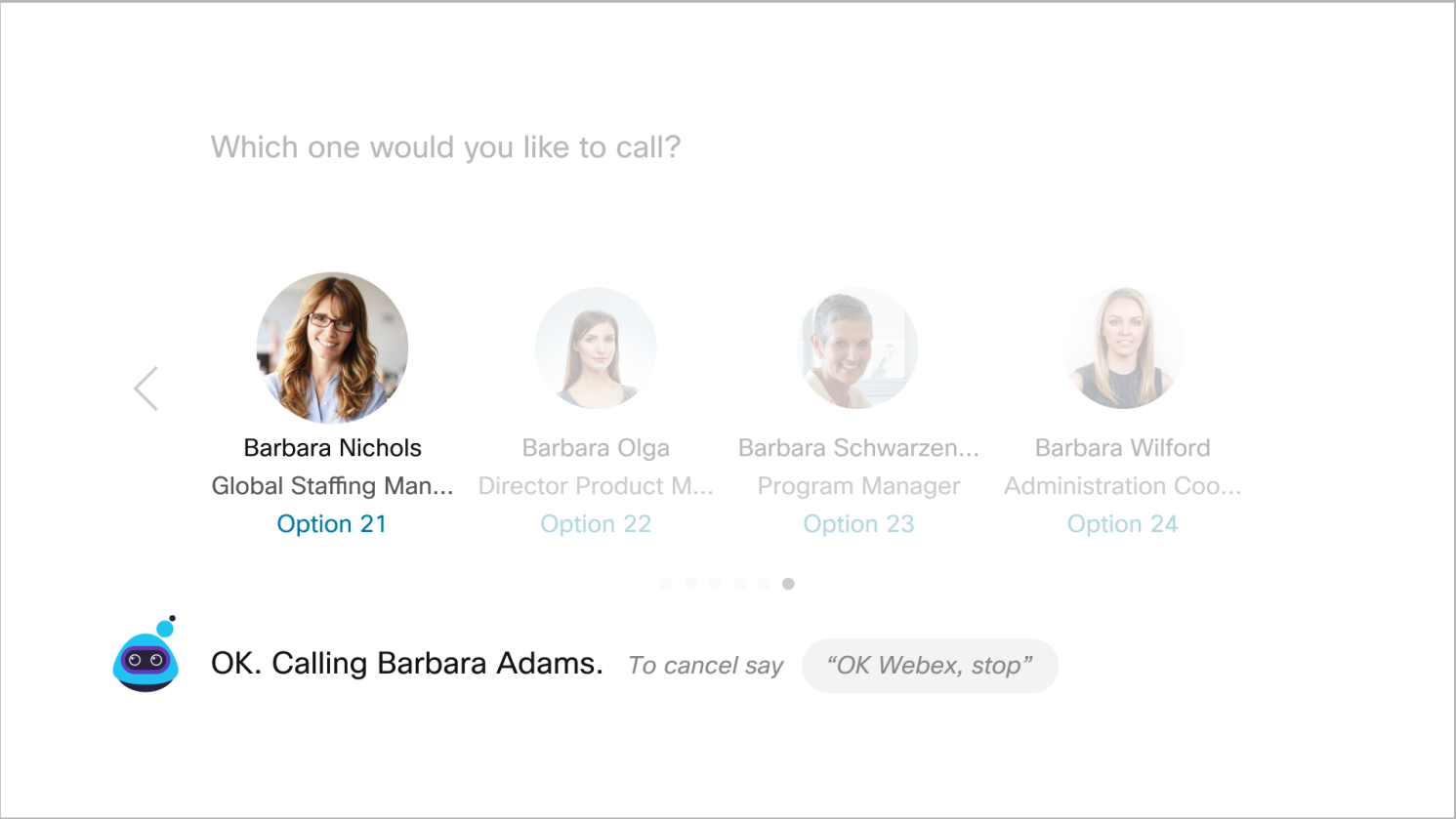
For Webex Assistant, we always start hints with "try saying." It feels more like a suggestion than a demand, and by keeping the wording and UI style consistent, we help users spot hints easily.

For each command example, we suggest only including what's after the wake word. In the 'call by name' example above, we displayed "option 1" rather than “OK Webex, option 1.” This is because most hints are shown when the assistant is listening, so there's no need for the user to say the wake word again.

On the second page of the carousel, we show “option 5” and “go back.” It wouldn't make sense to keep “option 1” since that option is no longer visible. Displaying “go back” instead of “show more” helps the user learn how to return to the previous screen.

We recommend that you leverage hints in this same way. Update your hints to be relevant to what your skill is displaying or what flow it's currently in. Suggested actions, next steps, and navigation tips are all great ways to use hints.

If the assistant isn't going to be in listening mode, you should include the wake word within the hint.



Outro

When the user exits the skill, we recommend showing an outro message. An outro helps to acknowledge what the user has said and reminds them to use the skill invocation phrase to interact with your skill again.

There are two scenarios where you should consider adding an outro message:



1. The user explicitly asked to exit the skill

Similar to how a user can say “OK Webex, dismiss” to make the assistant go away, you should allow your users to dismiss your skill as they wish. When they do, we suggest using an outro message like ***Thanks for using <skill_name>!*** Followed by a hint like ***To use this skill again, just say "OK Webex, ask <skill_name>..."***

The first half of this outro makes the experience more conversational, while the second half helps make it clear to the user that if they want to ask the skill for something, they'll have to start the interaction again. This helps to train users on the difference between interacting with the regular Webex Assistant and your skill.

2. Your skill is going to stop listening and dismiss itself

In some cases, when the conversation ends, your skill might stop listening and go away after giving a response. For example:

 **User:** “OK Webex, ask Weather, what’s the weather in San Francisco?”
 **Weather skill:** “Today in San Francisco, it’s 73 degrees and sunny!”
To use this skill again, just say “OK Webex, ask <skill_name>...”

“Today in San Francisco, it's 73 degrees and sunny!" will be a verbally spoken response. The hint that's displayed following the response indicates to the user that the skill will now dismiss itself.

Writing Natural Language Responses

So far, we’ve explored the high-level structure of conversations. This section will provide some specific tips for the replies that make up those conversations.

When users talk with your skill, they’ll ask for something and your skill will reply. The replies that the user hears from your skill are called Natural Language Responses (NLRs). These responses aren’t just a receipt of an action that has taken place—they’re a core part of the experience.

The responses should be efficient, natural, and consistent to achieve the best user experience. In this section, we’ll make recommendations for the length, flow, tone, and word choice of your skill’s replies.

NLR Length

Responses should be as brief as possible, without sacrificing naturalness or causing the tone to sound clipped or brusque. You should communicate the information that the user needs—no more and no less. Ideally, you should be able to say your response in one breath.

Brevity in your responses is important because it makes the information easier for users to understand, and there’s a limited amount of space on-screen. To avoid overflow in the UI, responses shouldn't be longer than 130 characters.

Conversational Flow



In natural conversation, we use words that mark the conversation flow, which makes it easier for the listener to follow along. Your Webex Assistant Skill should do the same. This includes markers to affirm what the user has said, mark the steps the user needs to take, or indicate a difference between your skill and the user's expectations.

1. Use an affirmation when the NLR confirms a response

Using an affirmation at the beginning of the reply lets the user know that your skill understood their request and is continuing the conversation based on the information from the user.

Affirmation markers: “ok,” “got it,” “sure,” “sure thing”

 *User: “OK Webex, ask Weather to show San Francisco forecast”*

  *Weather skill: “Here is the weather”*



  *Weather skill: “OK! Here’s the weather”*



2. Use timeline markers when explaining steps

When instructing the user, it's helpful to use timeline markers to guide them through steps or actions that they need to take, or to add continuity when giving the user a lot of information. When users know how much information is coming, it helps them to better take in that information.

Timeline markers: “first,” “next,” “then,” “finally”

 *User: “OK Webex, ask Weather for next week’s forecast”*

  *Weather skill: “Here is the forecast: Monday and Tuesday are going to be warm. Wednesday is going to cool down some and there might be rain. Thursday is going to warm back up. On Friday it will rain again.”*

  *Weather skill: “Here’s the forecast: Early in the week, it’s going to be warm. Then, it’ll cool down and there might be some rain Wednesday. Thursday heats back up and finally, on Friday we’ll be back to rain.”*

3. Use “Hmmm” to indicate a mismatch in user expectations

This marker serves as the opposite of an affirmation marker. It lets users know that there is some misunderstanding between your skill and what the user is asking. This can occur when the system isn’t able to do something the user asks,

or the users asks for something that doesn’t match with the state your skill is in. Adding this marker can also make your skill sound less brusque.

 *User: “OK Webex, ask Weather to close the web view”*

  *Weather skill: “The web view isn’t open.”*

  *Weather skill: “Hmmm, there isn’t a web view open right now”*

Tone

Your skill’s tone should be friendly, professional, and polite. Webex Assistant and its skills are part of an enterprise assistant, so be sure to make your tone friendly, but not overly casual. Think of the tone as business casual.

Here’s an example for a greeting:

  *Too casual “Hey! What’s new in your life today?”*

  *Too formal “Greetings.”*

  *Just right “How are you this afternoon?”*

While the first example is friendly, it might be overly so. The second is a formal greeting and sounds a bit terse. Asking a simple question strikes the correct balance here.

Here’s an example for asking the user to choose an option:

✗ 🤖 *Too casual* “Pick an option.”

✗ 🤖 *Too formal* “You can pick someone from these options.”

✓ 🤖 *Just right* “Which option would you like?”

Phrasing the reply as a question sounds much more natural than commanding the user to make a selection. Questions can help users feel like they’re having a conversation with your skill, rather than the skill just needing to get information from them.

Word Choice

When using a Webex Assistant Skill, users should feel like they’re talking with a person, not a robot. Colloquial language helps the responses feel more natural. When writing replies for your skill, think about how you would talk when chatting with a non-technical friend.

Here are some ways you can make your replies feel more natural:

1. Use contractions

Contractions are an easy way to make your replies feel more natural. It’s a small change, but one that makes a difference when hearing the spoken reply.

✗ 🤖 “I can not help you with that right now”

✓ 🤖 “I can’t help you with that right now”

2. Use pronouns

Use pronouns like “this,” “that,” “here’s,” and “it” to refer to nouns you’ve already mentioned. If you repeat the noun over and over, it begins to sound strange. If your skill mentions the same entity multiple times, we recommend making the subsequent mentions pronouns.

💡 *User: “I’d like to book **this room**”*

✗ 🤖 “OK! I can book **this room** for you. Would you like me to book **this room** for you?”

✓ 🤖 “OK! I can book **this room** for you. Would you like me to book **it**?”

3. Avoid technical language and technical abbreviations

We recommend leaving abbreviations and jargon out of your skill’s reply. Users don’t know, and probably don’t want or need to know, specific details. Just share what impacts them, like whether they’re able to use a certain feature.

✗ 🤖 *“PSTN calling is not supported on this device.”*

✓ 🤖 *“Dialling phone numbers isn’t currently supported on this device.”*

4. Use “any” instead of “no” in negated sentences

Negated sentences with “no” are correct, but sound more formulaic, like reading logic. We recommend using contracted negation rather than “no.”

✗ 🤖 *“There is no incoming call”*

✓ 🤖 *“There isn’t an incoming call to answer”*

5. Use “I” when your skill completes an action

If your skill performed an action, use a response that uses “I” as the subject to show that your skill is the one who has completed the action. This gives the user the feeling that your skill has succeeded, which builds credibility.

✗ 🤖 *“There are 2 meetings.”*

✓ 🤖 *“I found 2 meetings.”*

Error Handling

There will be times when your skill won’t be able to handle everything, and that’s okay. To give your users the best experience, we recommend having various error messages and opportunities for the user to try again. This helps users understand what’s happening and, ideally, allows them to recover from the error.

1. Skill Error

Issue: The user asks for something that your skill can do, but the skill can't complete the request.

Sometimes your skill might encounter an issue. Let your user know that your skill is experiencing an issue processing requests and that they should try again later.

💡 *User: “OK Webex, ask Weather to check the temperature”*

🤖 *Skill: “Hmmm. Something went wrong. Try again later.”*

2. Unknown Request

Issue: The user asks for something your skill doesn’t support.

When a user asks for something that your skill can't do, you can use a generic error message that lets the user know that feature isn’t supported.

For example, the Weather skill supports checking the temperature and the forecast but can't check the air quality:

🗣️ *User: “OK Webex, ask Weather to check the air quality”*

🤖 *Skill: “I’m not able to help with that just yet!”*

If you notice a pattern in the unsupported queries that your users are attempting, you can collect them into an intent and give a more specific error message. You can also remind the user what you’re able to help with if it’s relevant to their query. Taking the example of air quality with our Weather skill, you might choose to reply with something more specific like:

🗣️ *User: “OK Webex, ask Weather to check air quality”*

🤖 *Skill: “I’m not able to check the air quality, but I can tell you today’s forecast”* ”

3. Unknown Entity

Issue: The user asks for something your skill supports, but your skill can’t resolve the entity or there are no results for that entity.

When your skill isn’t able to resolve an entity, let the user know by repeating the entity in the error message (with automatic speech recognition errors, it can be helpful for the user to see the query) and let them know the skill didn’t find any results.

🗣️ *User: “OK Webex, ask Weather to check the weather in Eventide”*

🤖 *Skill: “There’s no weather information for Eventide”*

4. Unknown User Response

Issue: Your skill is listening for a reply from the user, but the response isn’t something that your skill expects as an input.

Sometimes when a skill is listening, it might capture background speech or the user’s request might transcribe imperfectly. Rather than ending the interaction, we recommend giving the user another opportunity to reply with the relevant information.

🗣️ *User: “OK Webex, ask Weather to check the forecast”*

🤖 *Skill: “Where would you like me to check the weather?”*

🗣️ *User: “Oh yeah, I loved that movie”*

🤖 *Skill: “Hmmm, where should I check the weather?”*

🗣️ *User: “San Francisco”*

5. Interaction Error: No User Response

Issue: Your skill is listening for a reply from the user, but the user doesn’t respond.

Sometimes when a skill is listening, it might capture background speech or the user's request might transcribe imperfectly. Rather than ending the interaction, we recommend giving the user another opportunity to reply with the relevant information.

🗣️ *User: "OK Webex, ask Weather to check the forecast"*

🤖 *Skill: "Where would you like me to check the weather?"*

🗣️ *User: "..."*

🤖 *Skill: "Hmmm, where should I check the weather?"*

For these multi-turn re-prompts, we recommend that you limit the number of re-prompts. We only allow Webex Assistant to re-prompt three times. This avoids cases where the assistant could get stuck in a re-prompt loop as it picks up background noise, or the case when the assistant isn't able to understand the user and we don't want to continue to frustrate them. After exhausting the re-tries, we let users know that we weren't able to understand them.

🤖 *Skill: "Hmmm, I wasn't able to understand how to help."*

Other Error Handling Guidelines

Here are some best practices for wording error messages.

1. Avoid Apologies

There's no need to apologize for not finding results or for not yet supporting a feature.

2. Tell the User Why

Make sure that your error message makes clear what went wrong. Let's look at an example error message:

🗣️ *User: "OK Webex, ask Weather what the temperature is here"*

❌ 🤖 *Skill: "I'm unable to access the geography code for this device"*

This error message communicates that the skill can't access certain information, but it doesn't explain why this relates to the user's request. The user asked for the temperature, but we're telling them something about the device instead. The user might infer that the geography code is needed to check the temperature, but, ideally, we prefer to be as specific as possible.

Here's a better way to phrase this error message:

🗣️ *User: “OK Webex, ask Weather what the temperature is here”*

✓ 🤖 *Skill: “Hmmm, I’m not able to get the temperature because I’m not sure where you are.”*

This response lets the user know the root cause of the issue and why the skill can't complete the request.

3. Avoid Using “I” in Error Messages

Avoid using “I” when your skill can't perform a request, specifically when your skill isn't the reason for the error. Removing “I” pronouns and using sentences with “there” for cases where the response is negative boosts the user's trust in your skill.

🗣️ *User: “OK Webex, ask Weather to get the weather in Eventide”*

✗ 🤖 *Skill: “I didn’t find weather information for Eventide”*

✓ 🤖 *Skill: “There isn’t any weather information for Eventide”*