German Autolabs

# How to write for voice assistants

2

# How To Write For Voice UI

When writing prompts for the VUI, it might be helpful to recall our core voice and tone guidelines:

- → Efficient
- → Concise
- → Productive
- → Intelligent
- → Innovative

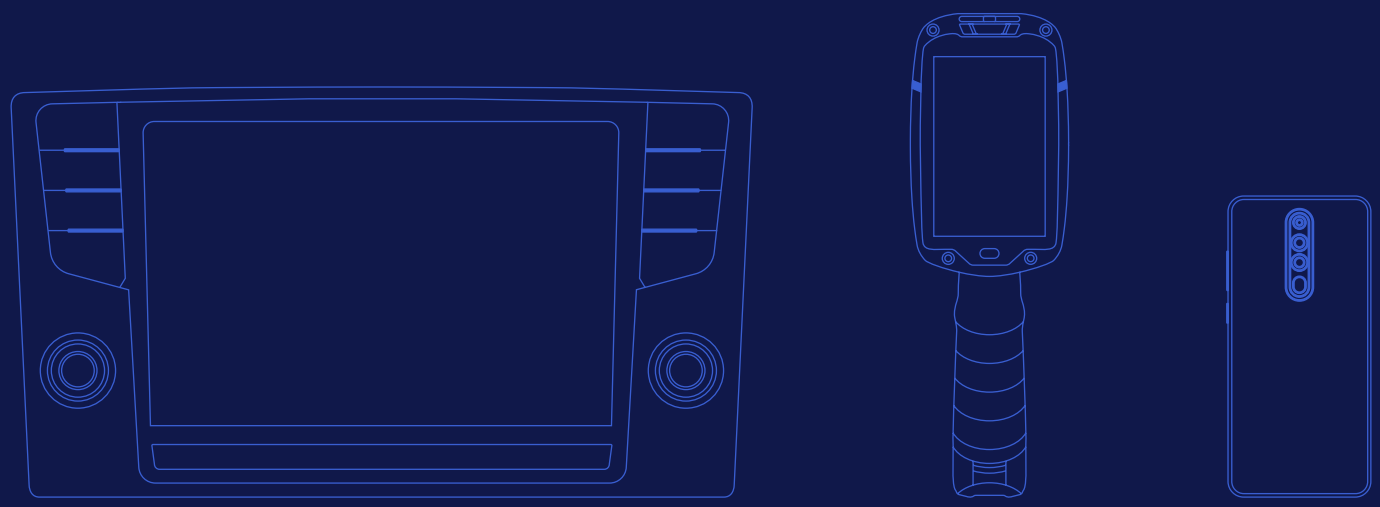
Voice UI is a very specific domain of conversation. We always try to follow these mantras when writing for our voice assistant:

**Be adaptable**
Let users speak in their own words.

**Be personal**
Individualize the entire interaction.

**Be available**
Collapse menus; make all options top-level.

**Be relatable**
Talk with them, not at them.



# 3 System and user personas

Part of the role of a conversation designer is that of a screenwriter. Before you can write a dialog, you have to have a clear picture of who the characters are. Personas are the design tool used for this. A good persona is specific enough to evoke a unique voice and personality, yet brief enough that it's easy to keep top-of-mind when writing a dialog. It should be easy to answer the question: "What would this persona say or do in this situation?"

## System persona

The system persona is the conversational partner created to be the front end of the technology that the user will interact with directly. Defining a clear system persona is vital to ensuring a consistent user experience. Otherwise, each designer will follow their own personal conversational style and the overall experience will feel disjointed. Persona creation typically starts with brainstorming adjectives (e.g., friendly, trustworthy) and narrowing them down to a short list. This list becomes a short description, often accompanied by images.
At German Autolabs, we've created the Chris system persona. Everything our assistant does (e.g., says, writes, displays, suggests) and everywhere Chris appears (e.g., the look and feel of the software and hardware) were designed to evoke a consistent persona.
Our German Autolabs system persona follows our brand guidelines and is: efficient, concise, productive, intelligent and innovative.

## User personas

Think of a few specific people you expect to use your assistant. Try to have 2-3 different types, e.g., a bus driver vs a traveling salesman. These user personas will help you avoid designing only for yourself and your goals.
Our German Autolabs personae currently include: last mile delivery drivers, bus drivers, CRM-access salespeople and motorhome owners.

# 4 Conversation for assistants

Conversation design is about teaching computers to be fluent in human conversation and its conventions.

### Start with what humans do

Conversations with a computer should not feel awkward or break patterns that have evolved over the past hundred thousand years. Instead, computers should adapt to the communication system users learned first and know best. This helps create an intuitive and frictionless experience.

### Adapt to technical limitations

In some ways, computers fall short of human capabilities. Technical limitations introduce scenarios that don't occur in human-to-human conversation. For example, human conversation never fails due to an unrecoverable error. Human conversation doesn't require starting with a specific word or phrase, e.g., "Hey Chris". In these cases, rely on user research to determine the best approach.

### Leverage technical strengths

In other ways, computers can exceed human capabilities. They don't get tired of being asked the same questions. They aren't offended by being given commands. There's no need to pepper their responses with filler words or other formulaic language, e.g., ums and ahs. They can quickly find and share information. Look for opportunities to avoid annoyance, streamline conversations, and exceed expectations.

# 5 Language

General style guide for spoken prompts.

### Focus on the user
Make the user the center of attention, not your persona. User-focused text keeps the conversation on track. It's more crisp and to-the-point.

### Be consistent
Try to keep the same logic patterns across different prompts. This helps the user to understand what the assistant can do and how they should interact with it.

### Don't launch into monologues
Be informative, but keep responses concise. Let users take their turn. Don't go into heavy-handed details unless the user will clearly benefit from it.

### Use short, simple words
Plain and simple language has the broadest appeal, making it accessible to people of all backgrounds.

### Avoid jargon
Be mindful of terminology, and watch for specialized expressions that can elicit misunderstanding and mistrust. Types of content that are especially prone to jargon include: sign-up flows, error handling prompts, and descriptions of settings.

### Randomize prompts when appropriate
Craft a variety of responses just like a person would. This makes the conversation feel more natural and keeps the experience from getting stale.

### Lead with benefits
If you want the user to do something, give them a reason first. Follow this formula: "To get what you want, do this thing."

### Avoid niceties
Niceties make responses feel distant and formal. Ditch them to keep the conversation friendly and informal.

### Use contractions
Spelling out words like "cannot" and "do not" can sound punishing and harsh.

### Don't provide UI-specific directions
Refer to actions and concepts instead. This keeps the information fresh and accurate even as the interaction design evolves.

# 6 Conversational components

### Acknowledgements
Okay.

### Apologies
Sorry, I can't send eCards yet.

### Commands
Navigating to Alexanderplatz.

### Confirmations
Got it. Did you mean Alexanderplatz, Berlin?

### Discourse markers
By the way, ...

### Earcons
<Welcome chime when German Autolabs powers on>

### Endings
Anything else I can help you with?

### Errors
Sorry, where would you like to go?

### Greetings
Welcome

### Informational statements
The weather outside is 13ºC. Your fuel level is at 80%.

### Questions
Where would you like to go?

### Suggestions
Try saying "Send a WhatsApp message" or "Play music"

# 7 Error handling

Even with robust intents, there is still room for error. Users may go off script by remaining silent (a No Input error) or saying something unexpected (a No Match error). Use error prompts to gently steer users back towards successful paths or reset their expectations about what is and isn't possible.

Good error handling is context-specific, so prompts for No Input and No Match errors must be designed for every turn in the dialog.