

ORACLE

Building Multilingual Digital Assistants

Program agenda

- 1 About multilingual digital assistants
- 2 Translation services
- 3 Native multilingual NLU
- 4 Resource bundles
- 5 Building multilingual digital assistants
- 5 Additional considerations

Program agenda

- 1 About multilingual digital assistants**
- 2 Translation services
- 3 Native multilingual NLU
- 4 Resource bundles
- 5 Building multilingual digital assistants
- 5 Additional considerations

What developers are looking for

One digital assistant that serves multiple languages

Develop in native language

Handle regional differences

All data should stay in Oracle cloud



Multilingual support in Oracle Digital Assistant

Translation services

Use Oracle, Google or Microsoft translation services to translate user language to English and, optional, bot responses to user language

- Recommendation is to use resource bundles for bot responses

Detects user language

Support for a large set of languages

Native multilingual NLU

Smaller set of supported languages

Pre-trained multi language model

- Detects & understands languages

Required resource bundles for outgoing messages

”

Choosing a translation option is a first design decision you make. If you can, Native multilingual NLU is recommended.

Program agenda

- 1 About multilingual digital assistants
- 2 **Translation services**
- 3 Native multilingual NLU
- 4 Resource bundles
- 5 Building multilingual digital assistants
- 5 Additional considerations

Translation service configuration

Translation services from Oracle, Microsoft or Google are configured on the ODA instance

- Settings | Translation Services

Makes translation service available but does not automatically apply it to a skill

The screenshot displays the Oracle Cloud console interface for configuring translation services. On the left is a dark sidebar with navigation options: Home, Development, Analytics, Settings, Audit Trail, Authentication Services, Data Management, Feature Management, Translation Services, API Services, and Additional Services. The main content area is titled 'Settings • Translation Services' and shows a list of services under the heading 'Services (1)'. A '+ Add Service' button is visible above the list. One service, 'Google', is listed and highlighted in light green, with a trash icon to its right. To the right of the list is a configuration panel for the 'Google' service. It includes a 'Service Type' dropdown set to 'Google', a 'Base URL' text input containing 'https://translation.googleapis.com/language/translate/v2', and an 'Authorization Key' text input with a masked key represented by a series of dots.



Skill configuration

The **Translation Service** setting is only available for skills created with a **Primary Language** from the translation service list

Create Skill

Display Name
Will be used in the Skill Catalog. This value can't be modified later. Required

Name
The official identifier for the skill. This value can't be modified later. Required

Skill Version: 1.0 Platform Version: 22.10 (Latest)

Dialog Mode
 YAML Visual

One-Sentence Description

Primary Language (Translation Service)
Portuguese

Create

Settings

General Configuration Digital Assistant Events Q&A Routing Config

Language

Language Mode
Translation Service

Primary Language
English

Translation Service
None
Google

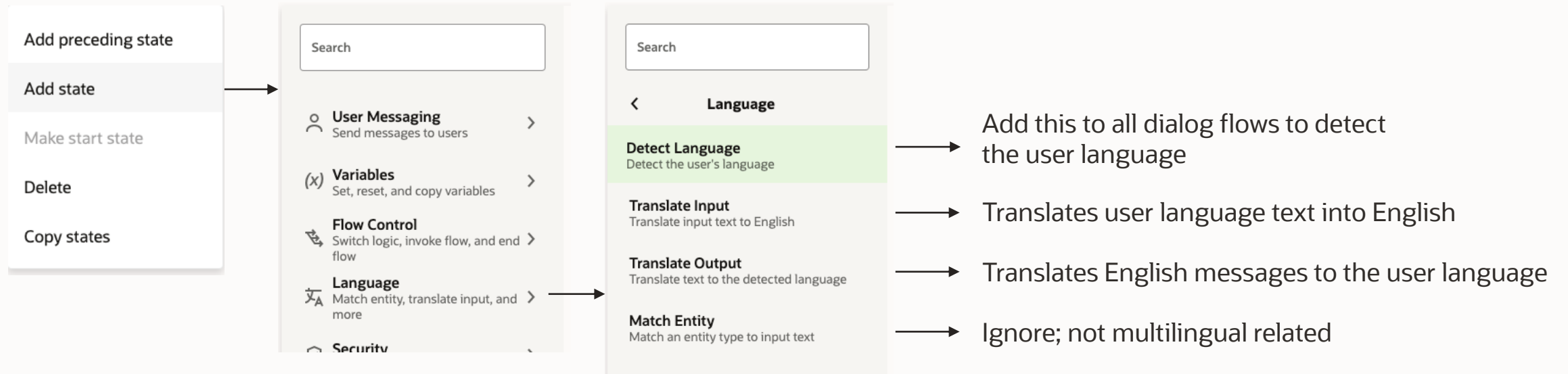


Flow configuration

Three component state templates are provided to work with translation services

"Detect Language" is the template that sets the profile.languageTag

- Add to all flows
- Ensures language specific resource bundles are used



”

Using a translation service to build multilingual skills, does not require changes to intents or entities

Program agenda

- 1 About multilingual digital assistants
- 2 Translation services
- 3 **Native multilingual NLU**
- 4 Resource bundles
- 5 Building multilingual digital assistants
- 5 Additional considerations

Getting started building native NLU multilingual skills

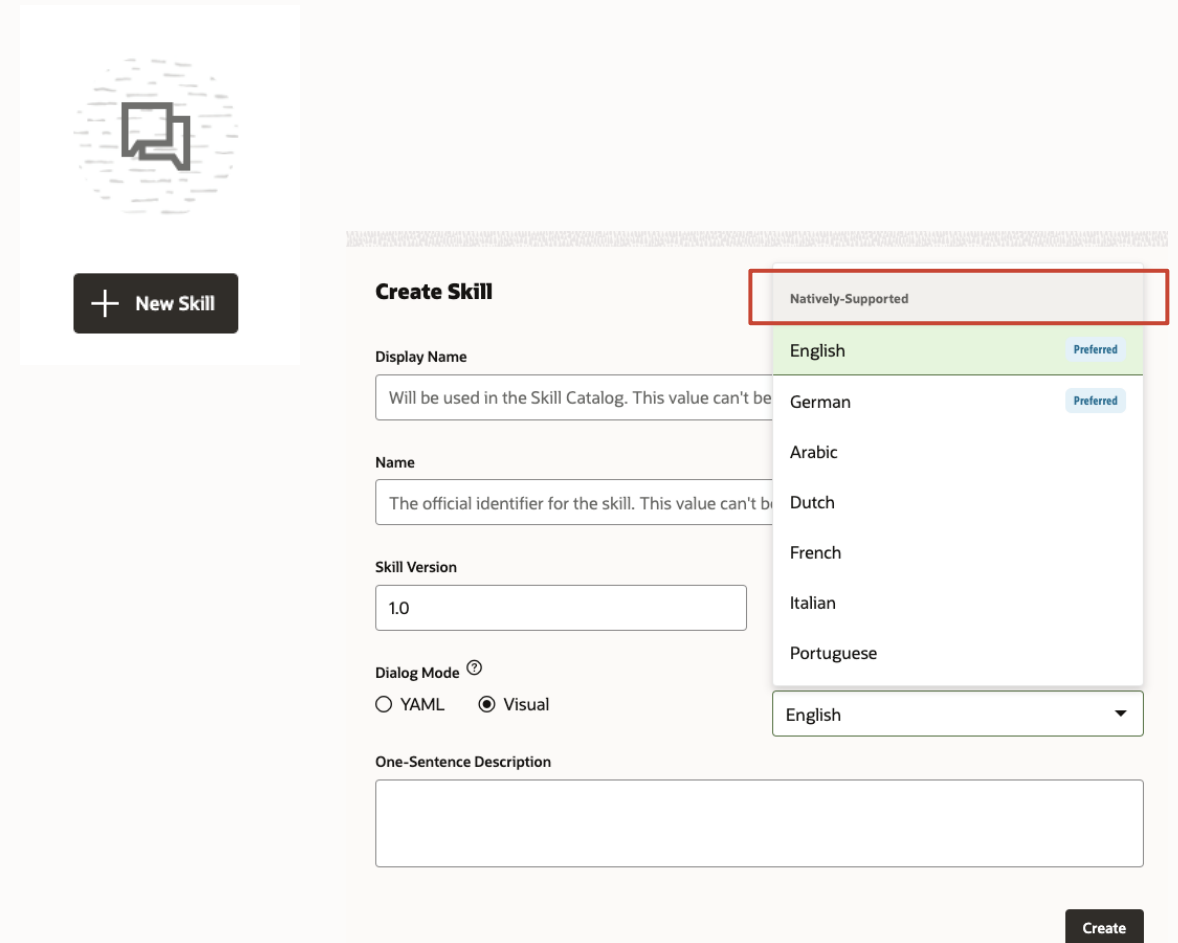
Natively supported languages are the default setting when building new skills

You need to select the primary language when creating a new skill

Because language support is built into the intent engine, native language support is superior to using a translation service

- No 3rd party dependencies

Number of supported languages smaller than with translation service



How native NLU language support works

How ML models work

Encodes text into high dimensional vectors

- Array of numbers
- Each element becomes single point in vector space

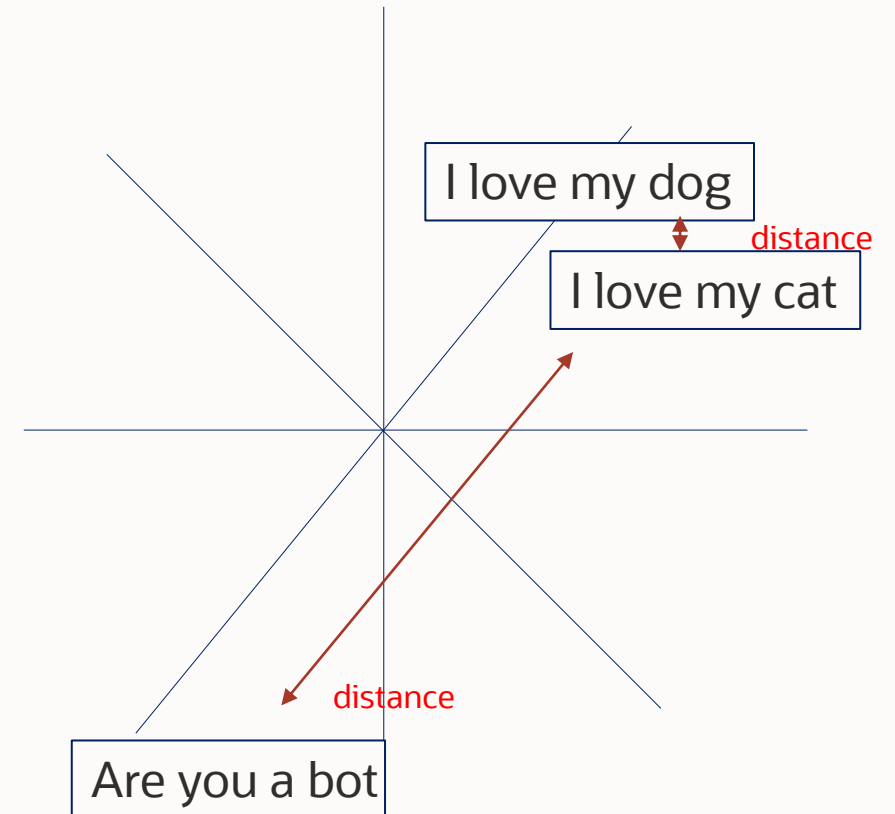
Embeddings

- "lookups" allowing to track similarities and relations

Pre-trained models

- You "refine" model by adding training utterances

high-dimensional vector



NLU native language support

Model maps sentences of multiple languages into single vector space

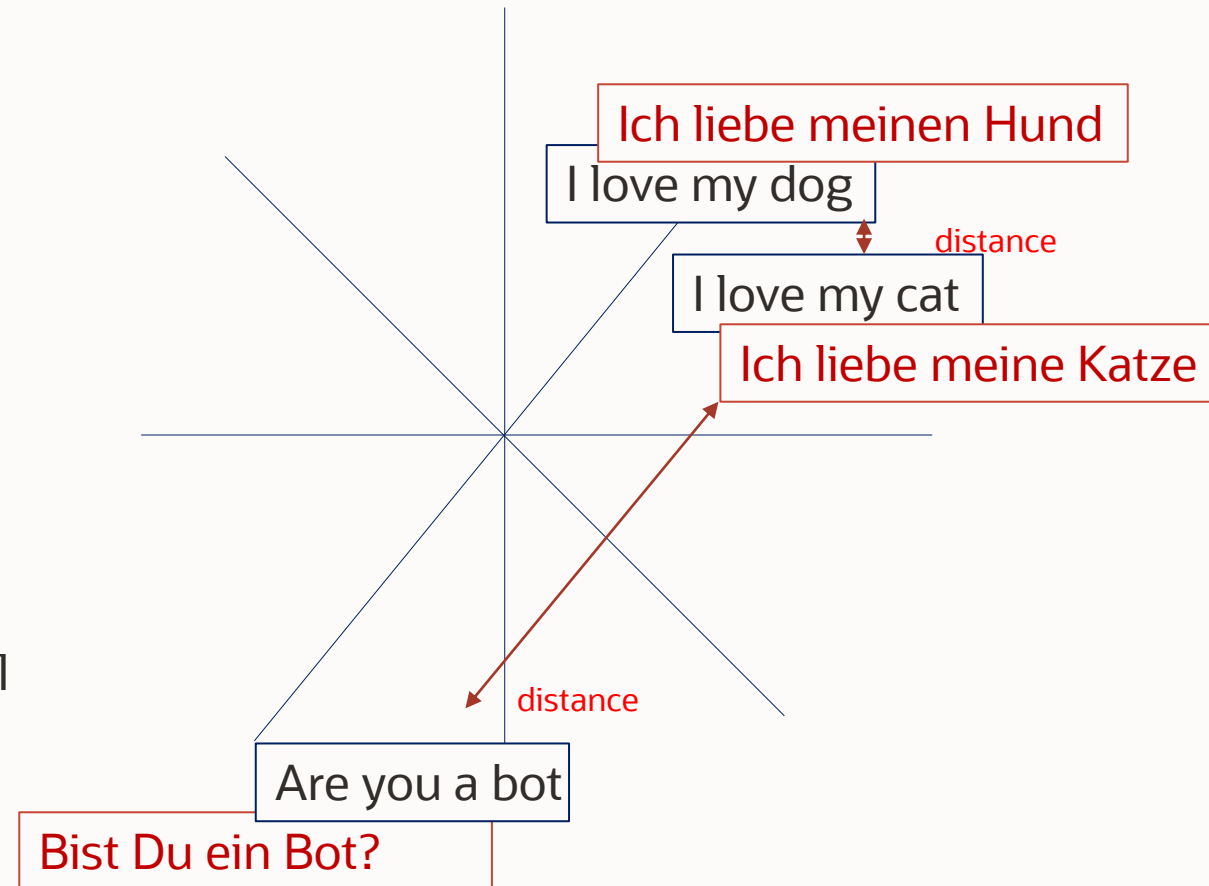
The closer two sentences are in the vector space, the more likely it is that they match

- Semantic similarity

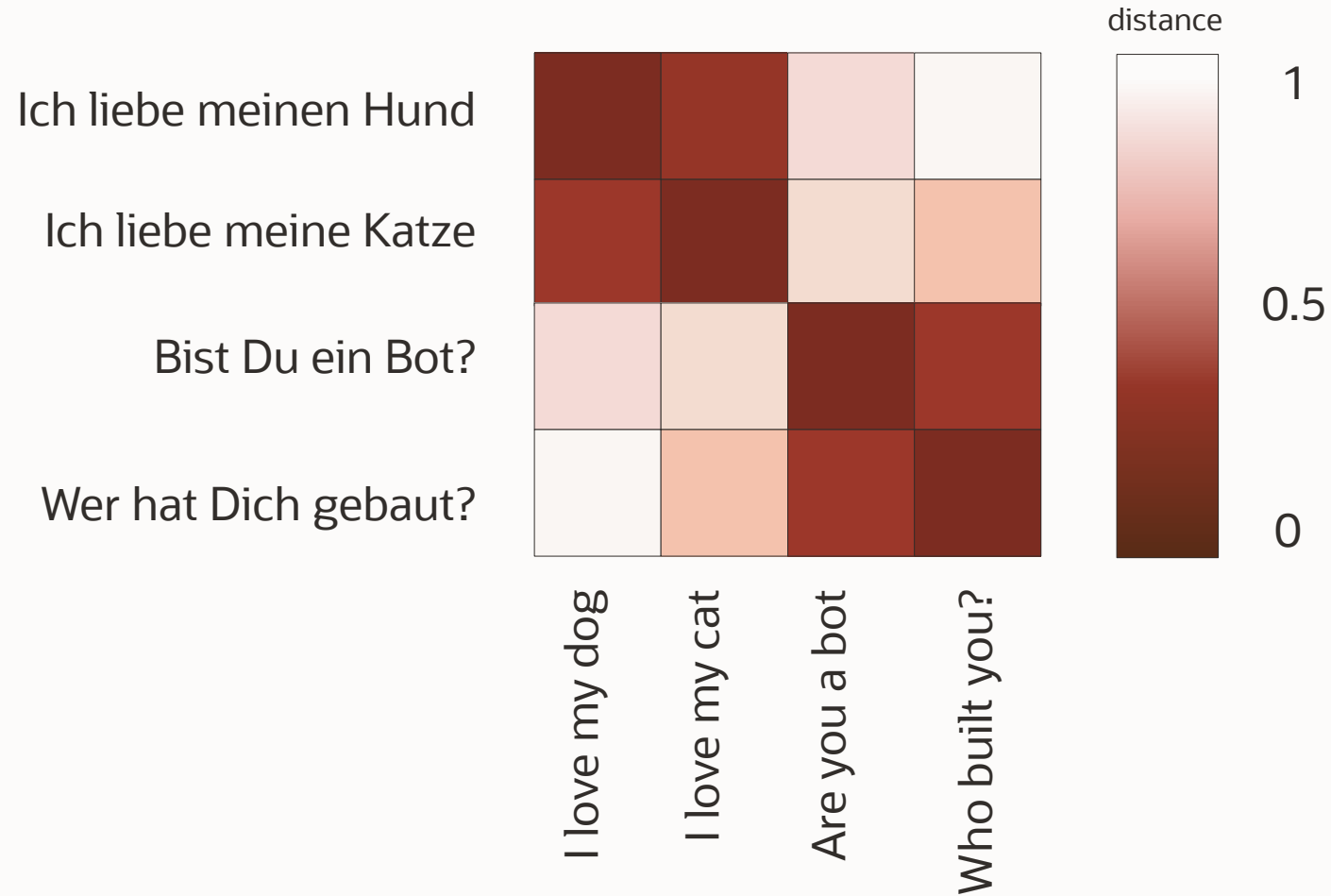
Pre-trained models

You can train the model on all languages you need

- Assuming the language is supported by the model



Simplified multilingual similarity map



One more example, and then we should be good

User says

Ich habe mir einen Hund gekauft (I bought a dog)

Model understands (high dimensional vector)

0.2	1.3	1.5	0.3	2.1	4.9
-----	-----	-----	-----	-----	-----	-----	-----	-----

Model finds nearest distance to

I love my dog

Bot responds with answer to detected intent

Ich helfe Ihnen eine Tierversicherung abzuschließen (I will help you to get a pet insurance)

How to enable native NLU language support

Native multilingual NLU

Intent model is multilanguage enabled

- Understands user input in multiple languages
- **zero-shot** and **few-shots** training

Detects language at runtime

Does not translate outgoing messages

- Requires resource bundles

The screenshot shows the 'Create Skill' form with a dropdown menu open for selecting a language. The form fields are:

- Display Name:** Will be used in the Skill Catalog. This value can't be...
- Name:** The official identifier for the skill. This value can't be...
- Skill Version:** 1.0
- Dialog Mode:** YAML Visual
- One-Sentence Description:** (Empty text area)

The language selection dropdown is titled 'Natively-Supported' and lists the following options:

- English (Preferred)
- German (Preferred)
- Arabic
- Dutch
- French
- Italian
- Portuguese

The current selection is 'English'. A 'Create' button is located at the bottom right of the form.



Multilingual intent training

Intent is trained in the base language

Zero-shot training can understand secondary languages without additional training data

Few-shots adds additional training data which will improve the 'understanding'

The screenshot displays the Oracle Digital Assistant interface for managing intents. At the top, there are tabs for 'Intents (11)' and 'Test Utterances'. Below these are controls for adding intents, filtering, and sorting. A list of intents is shown, with 'smalltalk.reg.whoBuiltYou' highlighted. To the right, there are panels for 'Examples (70)' in English, Portuguese, and Spanish, each with an 'Utterances to Add' field. Below these are two 'Utterance Tester' windows. The first window shows the utterance 'quem te construiu?' and the results for the intent 'smalltalk.reg.whoBuiltYou' (87.6%) and 'unresolvedIntent' (47.0%). The second window shows the utterance 'como fuiste creado?' and the results for 'smalltalk.reg.whoBuiltYou' (86.2%) and 'unresolvedIntent' (46.5%).

Correctly classifies the intent for both secondary languages



”

Create second language utterances that complement the primary language

Don't just use machine translation services

Involve native speakers

Add 20 – 30 % of the number of utterances created for the primary language

How to translate entities

Entity translation

Built-in entities support native languages

Value list and dynamic list custom entities need to be translated for each language to support

- Translate entity values if you want to display values in the user language
- Add any list of synonyms that makes sense for a language

At runtime

I want a pizza

quero uma pizza

What is the pizza size?

- large
- small
- medium

Qual o tamanho da pizza?

- grande
- pequeno
- medio

English Portuguese Spanish

+ Value

Value (Primary Language)	Synonyms
large	l, lg, big
medium	M
small	s

Create Value (in Portuguese)

Primary Language Value *

Value

Synonyms

Create

English Portuguese Spanish

+ Value

Sort By Primary Language Value Ascending

Primary Language Value	Value	Synonyms
large	grande	L
medium	medio	m
small	pequeno	s



Program agenda

- 1 About multilingual digital assistants
- 2 Translation services
- 3 Native multilingual NLU
- 4 **Resource bundles**
- 5 Building multilingual digital assistants
- 5 Additional considerations

Resource bundles

” Use resource bundles for all outgoing messages and prompts

Even when using translation services, stay in control of how all outgoing foreign language messages

At runtime, the proper language message will be used from the resource bundle

systemAnswer_conversational.reg.greeting

+ Add Language

Language	Message
default	Hello there!
es (Spanish)	Hola, ¿qué tal?
pt (Portuguese)	Olá, tudo bem?

Answer intent – resource bundle

size

+ Add Language

Language	Message
default	What is the pizza size?
es (Spanish)	¿Cuál es el tamaño de la pizza?
pt (Portuguese)	Qual o tamanho da pizza?

Entity prompt - resource bundle

Program agenda

- 1 About multilingual digital assistants
- 2 Translation services
- 3 Native multilingual NLU
- 4 Resource bundles
- 5 **Building multilingual digital assistants**

NLU language support or translation service?

” Skills using NLU language support cannot be mixed with skills using a translation service in the same digital assistant

Either all skills use NLU language support, or all skills must use translation services

When using NLU language support, all skills must support the same primary and secondary languages

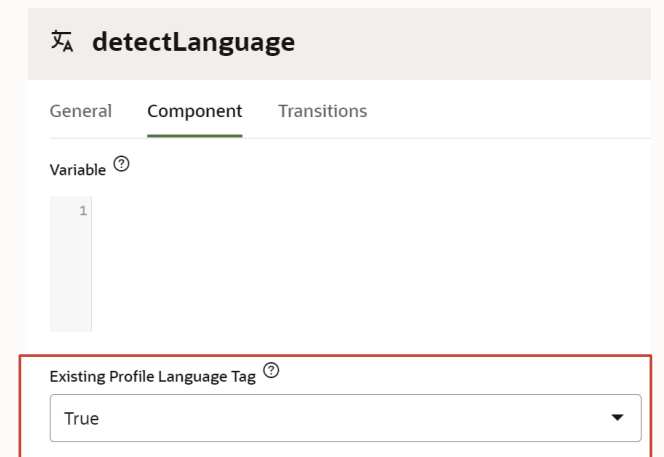
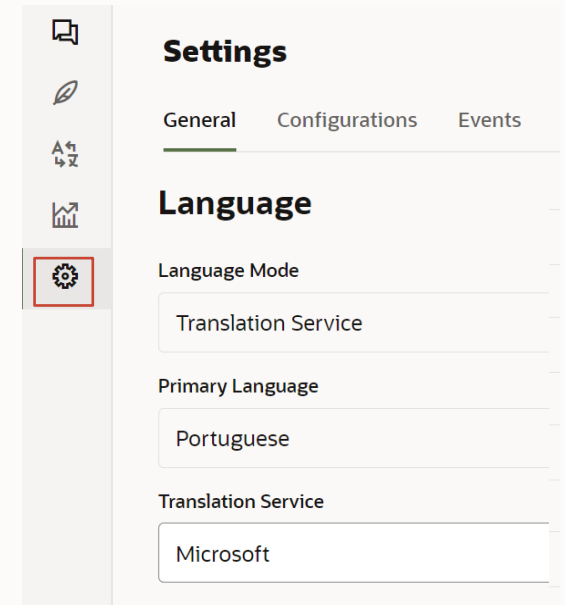
The choice of language translation is a design decision that you need to make early on

DA configuration

By default, if a digital assistant is set up with a translation service, the digital assistant will detect the user's language in the same way that a skill with the Detect Language component does.

The DA translates user input into English and passes that translated English text to the skill. Therefore, by default, the skill's Detect Language component would detect English as the language and set the **profile.LanguageTag** variable to English, even though the user entered non-English text.

To prevent that we need to set the Detect Language component's **Existing Profile Language Tag** property to **True** in each skill.



Resource bundles

” Make sure the Digital Assistant resource bundles are set to multilingual

The DA translation service can also translate the output, but it is advisable to use resource bundles, as it gives the business full control over the responses

This includes all the default resource bundles from the Digital Assistant, plus custom ones

The **One-sentence Description**, **Description**, **Invocation** fields and the **Sample Utterances** should also use resource bundles

Interaction Model

Invocation

Group

English ⓘ

Example Utterances (0)

`${rb('InsuranceSampleUtterance1')}`

`${rb('InsuranceSampleUtterance2')}`

`${rb('InsuranceSampleUtterance3')}`



System Intents

” Pre seeded data has data for all the natively supported languages

If you want to add your own system intent utterances, use resource bundle keys as well

The screenshot displays a configuration interface for system intents. On the left, a list of intents is shown: 'exit', 'help' (highlighted in green), and 'unresolvedIntent'. Below the list is a pagination control with a box containing the number '1'. On the right, the configuration for the 'help' intent is shown. It includes a 'Name' field with the value 'help', a 'Description' field with the text 'This intent is matched when users enter a g...', and language selection tabs for 'English', 'Spanish', and 'Portuguese'. Below these are sections for 'Examples' and 'Utterances to Add'.

”

Use the `#{profile.languageTag}` expression in skills and digital assistant to access the two-letter code of the detected language

ORACLE