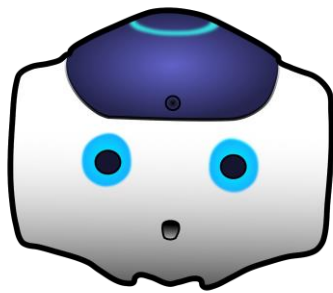


VERSION 1.0



Arigato

# NAO Documentation / User Manual

Human-Robot Interaction

CREATED BY CASEY CHEN, HAILEY DHANENS, MATTHEW HARKER,  
ANGIE QUACH, & DEREK VAUGHAN

ARIGATO, CENTRAL WASHINGTON UNIVERSITY  
ELLENSBURG, WA

“Our mission is to create and showcase meaningful and exciting human-to-robot interaction using the Aldebaran NAO robots recently required by CWU.” – AriGato Robotics

---

## CONTENTS

### Section 1: Introduction to the NAO Robot

1.1 What is NAO?.....	X
1.2 Who made NAO?.....	X
1.3 Definitions, Acronyms, & Abbreviations.....	X
1.4 References.....	X

### Section 2: NAO’s Capabilities

2.1 Initial (Pre-Built) Capabilities.....	X
2.2 Verbal Responses (Q&A).....	X
2.3 Internet-Based Requests & Responses.....	X
2.4 Facial Detection & Recognition Responses.....	X
2.5 Basic Movement Options.....	X
2.6 Advanced Movement Options.....	X

### Section 3: Creating Your Own NAO Modules

3.1 Basics of Choregraphe.....	X
3.2 Using the NAOqi Framework.....	X
3.3 Creating Your First Module.....	X
3.4 Integrating Your Module onto NAO.....	X

### Section 4: Help

4.1 General FAQs.....	X
4.2 Troubleshooting.....	X

## SECTION 1: INTRODUCTION TO THE NAO ROBOT

### 1.1 WHAT IS NAO?

NAO is an autonomous, humanoid, fully programmable robot. NAO robots are capable of 25 degrees of freedom, and thanks to their humanoid nature and design, are able to walk around, adapt, and interact with their surrounding environment. Furthermore, NAO has 4 directional microphones, loudspeakers, and 2 cameras capable of filming and analyzing the robot's environment, and human faces, for example. NAO is additionally capable of connecting to the internet by means of ethernet or Wi-Fi – this enables features such as http requests or big data analytics using the cloud.

All of these features add up to NAO's capabilities essentially being limitless, it is truly up to the developer's imagination to decide what the robot will eventually be capable of. Development for NAO can primarily be conducted in either Python or C++, though some other programming languages have small amounts of support as well (e.g. Java, MatLab).

Additional Specifications:

- Dimensions: 22.6 x 10.8 x 12.2 inches (574 x 311 x 275 mm)
- Weight: 12.08 pounds (5.48 kg)
- Autonomous Battery Life: 60 minutes active use, 90 minutes stationary use
- Operating System: Linux-Based NAOqi 2.8 (Linux Distro: Gentoo)
- Processor: Intel Atom E3845 @ 1.91 GHz

### 1.2 WHO MADE NAO?

The initial development of the NAO robot began as early as 2004. NAO was created by a French company known as Aldebaran, who was later acquired by SoftBank Robotics, a company based out of Japan, in 2015. The first public version of the NAO robot was released in 2008, however the version this project will be focusing on (NAO v6, or NAO Next Gen) was released to the public in 2014. Aldebaran also created the "Choregraphe" software that a bulk of the development of custom modules for NAO are made in.

### 1.3 DEFINITIONS, ACRONYMS, & ABBREVIATIONS

#### ALDEBARAN

A French robotics company, acquired by SoftBank Robotics in 2015. Developed NAO and Choregraphe.

#### API

Application Program Interface.

#### AUTONOMOUS LIFE

#### PLACEHOLDER TEXT

#### CHOREGRAPHE

A multi-platform desktop application that allows users to create animations and behaviors for the NAO unit, and test them in both simulated and real environments. It also allows users to monitor the NAO's visual and audio sensors as well as communication and error logs.

#### ETHERNET

A common form of network cable. It allows a connected device to join a local area network (LAN) in order to connect to and browse the internet.

#### LIBRARY

A collection of well-defined resources and implementations of behavior, written for/in a particular programming language for use by other developers to simplify and speed up development for a system.

#### MODULE

#### PLACEHOLDER TEXT

#### NAO

An autonomous, fully programmable, humanoid robot designed by Aldebaran Robots.

#### NAOQI

A Linux-based operating system stored in the robot's memory at all times; used for running and controlling features, programs, and modules.

#### SEE

The NAO robots cannot "see" in a physical sense but has cameras that it can use to record images to identify its surroundings.

#### SENSOR

Measures the robot's configurations, conditions, and its environment and sends such information to the robot for processing.

#### SOFTWARE

A set of computer instructions used to obtain input, and then manipulate that input in order to generate relevant output in terms of function and performance as specified by the user.

#### USER

A person who will interact with and make use of the NAO robot's various capabilities.

#### WI-FI

Short for "Wireless Fidelity". A means of allowing computers, smartphones, and other internet-enabled devices to communicate with one another wirelessly.

### 1.4 REFERENCES

- [1] NAO Lab Documentation/Info: <https://team.inria.fr/perception/demos/naolab/>
- [2] NAO, NAOqi, Choregraph Documentation: <http://doc.aldebaran.com/>

[3] NAO, Technical Guide: <http://doc.aldebaran.com/2-1/family/index.html>

[4] Engineering NYU, Intro to Robotics

<http://engineering.nyu.edu/mechatronics/smart/pdf/Intro2Robotics.pdf>

## SECTION 2: NAO'S CAPABILITIES

### 2.1 INITIAL (PRE-BUILT) CAPABILITIES

The original capabilities of the NAO are put on the robot by subscribing to the Aldebaran "Basic Channel." Official documentation on this channel lists and describes its capabilities.<sup>1</sup> Below is a small table listing some of the pre-built phrases that NAO can respond to, most of which are available in English, French, and Japanese Language settings.

"How are you?"	"Can you say goodbye?"	"What can you do?"
"Tell me all you can do."	"How do I install an application?"	"How do I start an application?"
"What did I say?"	"Can you repeat please?"	"What is your IP address?"
"Are you connected to Internet?"	"What languages so you speak?"	"Speak French."
"Can you speak French?"	"Speak Japanese."	"Speak Chinese"
"Can you speak Chinese"	"Speak softer"	"Speak louder"
"Can you stand up?"	"Can you sit down?"	"Crouch."
"Lay down"	"Lift your right arm"	"Lay down on your back"
"Lay down on your belly."	"Stop Looking at me"	"What is your name?"
"Introduce yourself"	"How much do you weigh?"	"How tall are you?"

### 2.2 VERBAL RESPONSES (Q&A)

#### DEVELOPERS

Verbal queues:

- "Who is developing your programs?"
- "Who is working on you?"

---

<sup>1</sup> [http://doc.aldebaran.com/2-1/nao/basic\\_channel\\_conversation.html](http://doc.aldebaran.com/2-1/nao/basic_channel_conversation.html)

- “Who is in the capstone project?”

Description: NAO will list the team members of the AriGato capstone project.

#### ***HOW OLD ARE YOU***

Verbal queues:

- “How old are you?”
- “What is your age?”

Description: NAO will respond with various humorous verbal responses.

#### ***SING THE ANTHEM***

Verbal queues:

- “Sing the anthem”
- “Anthem”

Description: NAO will begin to “sing” the American National Anthem (an .mp3 file is played over its loudspeakers) and place its hand over its heart.

#### ***WHAT TIME IS IT?***

Verbal queues:

Description:

## **2.3 INTERNET-BASED REQUESTS & RESPONSES**

#### ***MOVIE INFORMATION***

Verbal queues:

- “Give me Rotten Tomatoes ratings.”

Description:

#### ***TEMPERATURE***

Verbal queues:

- “What is the current temperature of Ellensburg?”
- “What’s the current temperature?”
- “What is the temperature?”
- “How hot is it outside?”
- “How cold is it outside?”
- “How hot is it?”
- “How cold is it?”

Description: NAO retrieves weather information from OpenWeatherMap.com and replies with the current temperature in Ellensburg Washington. Due to this module being intended for use in Ellensburg exclusively, if one wants to change the city, they will have to go into the module's code and change it manually.

## 2.4 FACIAL DETECTION & RECOGNITION RESPONSES

### *AGE GUESSER*

Verbal queues:

- "Can you guess how old I am?"
- "Guess my age."
- "How old do you think I am?"
- "What is my age?"

Description: NAO will use its facial detection and mapping abilities to attempt a guess at the user's age.

### *FACIAL RECOGNITION*

Verbal queues:

- "Do you know me?"
- "Do you remember me?"
- "Do you know my name?"
- "Do you know who I am?"

Description: Provided that a human face is within view - if NAO can recognize the face within 6 seconds, NAO will greet the person, if not, the module will time out, and NAO will say "Sorry, I do not recognize you".

**Note:** Learning new faces is currently only possible when the NAO unit is connected to a computer and the "Learn Face" box within Choregraphe is ran (i.e., there is no type of "Learn my face" command to have NAO learn a face on the fly).

## 2.5 BASIC MOVEMENT OPTIONS

### *JAZZ HANDS*

Verbal queues:

- "Jazz hands."
- "Do jazz hands."
- "Be jazzy."
- "Can you do jazz hands?"

Description: NAO performs "jazz hands" with its hands, essentially a small wrist motion.

### ***MOVE FINGERS***

Verbal queues:

- “NAO, wiggle your fingers.”
- “Move fingers.”
- “Can you wiggle your fingers.”
- “Do your fingers move.”
- “Move your fingers.”

Description: NAO will open and close its hands to demonstrate the mobility of its fingers.

### ***NOD YES***

Verbal queues:

- “Nod yes.”
- “Can you nod for me?”
- “Nod your head.”
- “Nod your head for me.”

Description: NAO will move its head up and down to demonstrate the mobility of its head/neck.

### ***RAISE LEFT/RIGHT FOOT***

Verbal queues:

- “Raise your left/right foot.”
- “Move your left/right foot.”

Description: NAO will lean to the side and begin to balance on one leg while lifting the corresponding foot in order to demonstrate the mobility of its legs and potential balancing capabilities.

### ***TURN HEAD LEFT/RIGHT***

Verbal queues:

- “Turn your head left/right.”
- “Turn head left/right.”
- “Head left/right.”
- “Move your head to the left/right.”
- “Look left/right.”

Description: NAO will turn its head to the corresponding direction in order to further demonstrate the mobility of its head/neck.



## *WALK FORWARD/BACKWARD/LEFT/RIGHT*

Verbal queues:

- “Walk <direction>.”
- “Move <direction>.”
- “Step <direction>.”
- “Take a step <direction>.”

Description: NAO will move 0.2 meters in the desired direction.

**Note:** Directional options include forward, backward, left, & right.

## 2.6 ADVANCED MOVEMENT OPTIONS

### *PUSHUPS*

Verbal queues:

- “Pushups.”
- “Do pushups.”
- “Do some pushups.”
- “Can you do pushups?”
- “Do you know how to do pushups?”

Description: NAO will ask how many pushups you would like to be performed (he can do between 1 and 10 inclusive), and will proceed to do that many pushups.

\*\*\*WARNING\*\*\*

NAO **must** have plenty of clear and empty space behind and in front of it to do pushups, and must be kept away from any ledges - otherwise damage to the NAO unit is highly possible. Furthermore, making NAO do excessive amounts of pushups without a break can lead to overheating motors and will cause NAO to shut down.

\*\*\*\*\*

## SECTION 3: CREATING YOUR OWN NAO MODULES

### 3.1 BASICS OF CHOREGRAPHE

### 3.2 USING THE NAOQI FRAMEWORK

### 3.3 CREATING YOUR FIRST MODULE

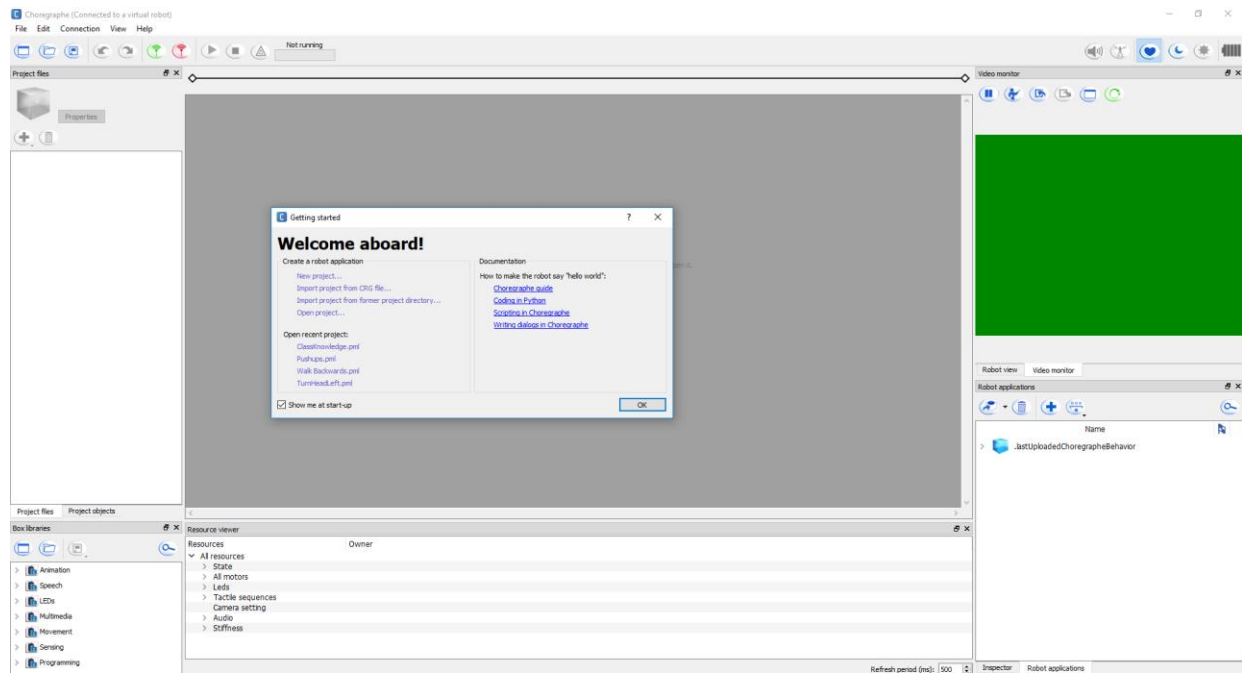


FIGURE 1 - FIRST OPENING CHOREGRAPHE

When first opening Choregraphe a window will pop up saying “Welcome aboard!” It will have a documentation section with a quick “Hello World” tutorial, and links to recent projects if there are any. To begin, either click “New project...” or exit out of the window, which will create a new project anyway. Start by filling out the information with the properties button in the upper left corner, next to the blue square shown in the left image of figure 2

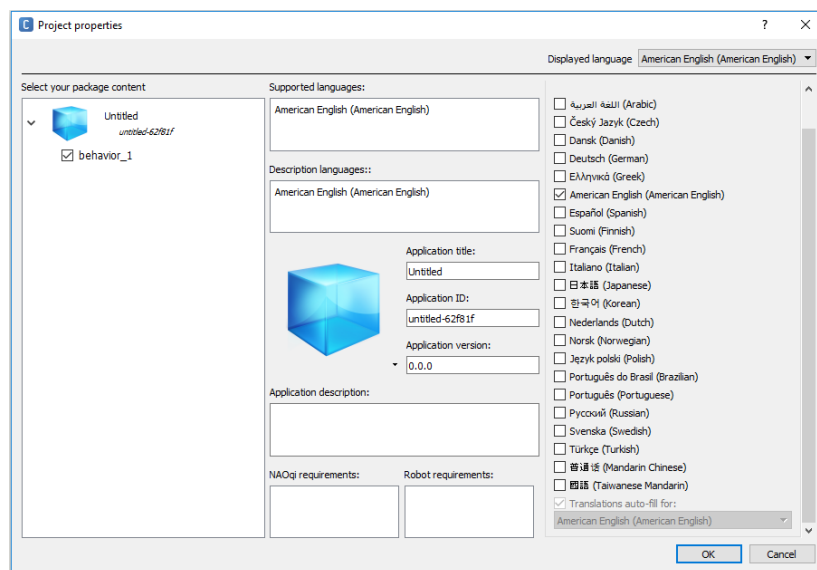


FIGURE 2 - PROJECT PROPERTIES

Fill in the “Application title” section, so that when you install your module on the robot you will know exactly which module it is (it will otherwise be untitled which can become quickly confusing). Next click on the text “behavior\_1” under the blue cube in the “Select your package content” section (upper left) in the window shown in figure 2.

Project properties

Displayed language: American English (American English)

Select your package content

- Example (untitled-62f81f)
  - ☒ behavior\_1

Name: behavior name

Nature: Interactive

Description:

☒ May start on user request

Trigger sentences:

Loading responses:

Permissions:

Launch trigger condition:

Behavior name: American English (American English)

OK Cancel

FIGURE 3 – BEHAVIOR PROPERTIES

Fill out the following sections shown in figure 3 after clicking “behavior\_1”:

- **Name:** Make sure your module’s name is self-documenting (i.e., The name should be relevant to what the module does). A good example for a simple “Hello World” module would be “hello”, and a bad example would be “module 1”, or “myModule”.
- **Description:** This section will let future users know what your module is supposed to do, and possibly why you have created the module (to serve some certain purpose, for example). It is best to have your description be straight to the point, and not too long – but also informative with all the relevant information for the module included.
- **Nature:** Nature has three possible settings: “Interactive”, “Solitary”, and “No Nature”. Make sure your module’s nature is set to Interactive. This means that a user can trigger the module with verbal trigger sentences while NAO’s autonomous life feature is on. Solitary means that NAO will perform the module when he is not being interacted with, and can be interrupted at

any time with Interactive modules. There is no available description for No Nature, so it is best not to make use of this mode.

- **Trigger sentences:** Trigger sentences are what NAO listens for to perform a corresponding module. Make sure each trigger sentence is unique, such that they don't share a phrase with another module that is already on the robot. It is also good to make a couple of similar trigger phrases for each module, so that the module can be more generally commanded.  
(Example: For an arm raising module, instead of just saying "lift arm", also put "lift YOUR arm", "Can you lift your arm" and "lift your arm please.")
- **Loading responses:** Loading responses are intended to be what the NAO unit will say after being told a trigger phrase, before the module is actually performed. However, in AriGato's experience, these loading responses do not actually work, and the robot only ever uses the default phrases of "Okay" and "Let's go!".
- **Permissions:**
- **Launch trigger condition:**

### 3.4 INTEGRATING YOUR MODULE ONTO NAO

## SECTION 4: HELP

### 4.1 GENERAL FAQs

### 4.2 TROUBLESHOOTING