Bridge Setup for Using Python 3 on NAO

Nao's software is outdated and still works with Python 2.7. However, sometimes, you want to use Python 3 code. For example, if there is a specific functionality you want to use that doesn't exist in Python 2.7, or if there is a deep learning model that uses/needs Python 3 that you want to incorporate. Jan de Wit made a solution for this problem (found in this git repo), that allows you to write your code in Python 3, and, through a bridge connection, converts it to the corresponding Python 2.7 code, which is then runnable on the NAO robot. Still, our group had some problems getting this bridge to work, so I made this document so that our group, and other groups in the future, know how to set up and use the bridge.

Basic setup

The following steps are basic setup and only have to be done once

1. **Fork and clone the repository**
    - Fork the git repository
    - Clone the repository to your laptop
2. **Install NAOqi SDK**
    - from https://www.aldebaran.com/en/support/nao-6/downloads-softwares/former-versions?os=45&category=76 download **Python 2.7 SDK setup**, and make sure you have **version 2.1.4** (bottom of the list)
3. **Install Python 2.7**:
    - Install Python 2.7 (32-bit version, for compatibility with the NAOqi SDK version we use)
4. **Create the Python 2.7 Executable**:
    - Download and install py2exe for Python 2.7 from https://sourceforge.net/projects/py2exe/
    - In the command line, go to `robot-jumpstarter-python3/python27`
    - Run `python setup.py py2exe` to make an executable (make sure to use your python 2.7 installation for this)
    - The executable (`naoqiconnection.exe`) and necessary libraries are placed automatically in this directory: `robot-jumpstarter-python3/python27/dist`
    - Py2exe also generates several .dll files, which are placed in the `robot-jumpstarter-python3/python27/dist` directory. You need to put these files in your local `C:/windows/system32` directory

Writing Python 3 code

Now, we can proceed with writing our code in Python 3

1. Write your code in Python 3, and store your scripts in `[git_repo_location]/robot-jumpstarter-python3/python3`
2. Use `ServiceCache` to call the necessary functions, the functions themselves should be the same as when you would use python 2.7 directly. The difference is that instead of instantiating a proxy directly, you should use the repository's `ServiceCache` to access a proxy and its functions.
    - **Example**: Normally, you would instantiate services directly with `ALProxy` in Python 2.7. **Below is an example of what you would do without the bridge. We will not be doing this:**

```
from naoqi import ALProxy

# instantiate text to speech proxy
tts = ALProxy("ALTextToSpeech", "192.168.1.2", 9559)

# use proxy to make function call
tts.say("Hello")
```

Instead, we do this: we use the `ServiceCache` (from the jumpstarter repository) to call the same function in Python 3. You can save your scripts in the `[git_repo_location]/robot-jumpstarter-python3/python3` directory. Or you can write your code somewhere else, but then you'll have to add `[git_repo_location]/robot-jumpstarter-python3/python3` to your PATH so that you can import the bridge and services.

```
import stk.python27bridge
import stk.services

# connect to the bridge (which should be running before you run
# this script)
bridge = stk.python27bridge.Python27Bridge()

# instantiate the service cache
sc = stk.services.ServiceCache(bridge)

# use the service cache to access the same tts proxy
tts = sc.ALTextToSpeech
# and execute the proxy's say function
tts.say("Hello")

# or do it in one line
# sc.ALTextToSpeech.say("Hello")
```

Starting the bridge

NOTE: you can either start the bridge manually every time, or use the `connection_setup.bat` script that is provided at the end of this document. This spares you the hassle of setting the python path and switching between choregraphe directory and your project files each time.

Either start the bridge manually

1. **Open terminal en set python path to naoqi SDK**
    - for example in powershell: `$env:PYTHONPATH = [path_to_sdk]` (probably something that ends with `site-packages/qi`)

2. **Open choregraphe via terminal**
    - `cd [choregraphe install location]/choregraphe suite 2.8`
    - run `./bin/choregraphe_launcher.exe`
3. **Connect to the robot**
    a. **virtual robot**
        - In choregraphe, select NAO in edit > preferences > virtual robot
        - connection > connect to virtual robot
        - check port: go to edit > preferences > virtual robot. At the bottom, it should state something like: "NAOqi state: Running on port [portnr]"
    b. **real NAO robot**
        - connect to the robot via the robot lab's router (TAs will explain how to do this)
        - press NAO's power button once to get its IP address (it will say it out loud)
        - By default, the robot should be using port number **9559**

4. **run python 2.7 executable to start bridge connection:**
    - in terminal, go to `[git_repo_location]/robot-jumpstarter-python3/python27/dist`
    - run `naoqiconnection.exe --qi-url=[IP address]:[portnr]`,
        - **Virtual robot:** `[IP address]` is your localhost IP address (`127.0.0.1`) and `[portnr]` is the port number you found in choregraphe (check port: go to edit > preferences > virtual robot. At the bottom, it should state something like: "NAOqi state: Running on port [portnr]")
        - **Real NAO robot:** `[IP address]` is the robot's IP address (press NAO's power button once) and `[portnr]` the port used for the connection (usually **9559**)

Or start the bridge with `connection_setup.bat` (see script below)

1. **(only once) change paths in `connection_setup.bat`**
    - **PYTHONPATH:** change the PYTHONPATH variable in the script to the path where you installed naoqi. Setting the path this way ensures that it sets the PYTHONPATH temporarily each time you run the script. This means that when your session is finished, PYTHONPATH refers to your usual python 3 installation again, and all your other programs that need that path are still functioning.
    - **Choregraphe path:** change the path so that it matches your choregraphe installation
    - **Project path:** change the path pointing to the dist folder inside your local bridge repository directory: `[path_to_repository]\robot-jumpstarter-python3\python27\dist`
2. **run `connection_setup.bat` in terminal**
    - locate directory where you put `connection_setup.bat`
    - simply type `connection_setup.bat` and hit enter
    - you will be asked to type `[IP address]:[portnr]`. To find `[IP address]` and `[portnr]`, see step 4 of 'starting the bridge manually'

Finally: run your python 3 scripts in new terminal session

- the bridge connection is running in your current terminal session, so you need to open another session to run your scripts
- go to `[path_to_repository]/robot-jumpstarter-python3/python3`
- Run the script you need, which contains functions that use the bridge
- The script automatically sends the commands to the Python 2.7 executable (`naoqiconnection.exe`), which then runs the python 2.7 code on the robot

The connection_setup.bat script for running the bridge (paths in yellow are what you need to change)

```
@echo off
echo Setting up the environment for the NAOqi Python SDK…

REM change the path below to your Python installation
set PYTHONPATH=D:\Python27\Lib\site-packages\qi

echo Starting Choregraphe…

REM change the path below to your Choregraphe installation
set CHOREGRAPHE_PATH=D:\Program Files (x86)\Softbank Robotics\Choregraphe Suite 2.8\bin
start "%CHOREGRAPHE_PATH%" choregraphe_launcher.exe

:waitForPort
SET /P PORT="Enter NAO's IP address and NAOqi port number separated by a colon (:) "

if "%PORT%"=="" (
      echo You must enter a port number.
      goto waitForPort
)

echo starting naoqiconnection.exe at port %PORT%...

REM change the path below to the dist folder in your clone of the repository
cd D:\Documents\UNI\Master\HRI\Project\robot-jumpstarter-python3\python27\dist

start naoqiconnection.exe --qi-url=%PORT% 2>&1
```