
pyoracle*forms*

Release 0.4.1

May 22, 2022

Contents

1	Installation	1
2	Working with a Forms Module	3
3	Reading and modifying properties	5
4	Iterating over Forms objects	7
5	Creating new objects	9
6	Using older Oracle Forms versions	11

CHAPTER 1

Installation

pyoracle_forms is available on PyPI and can be installed using pip:

```
pip install pyoracle_forms
```

The only external dependency is the Oracle Forms API, which you should have available as I assume you are working with Oracle Forms. If you do not, it is available on Oracle website here: [Oracle Forms](#)

Working with a Forms Module

First you must import the Module class and initialize the API.

```
>>> from pyoracle_forms import Module, initialize_context
>>> initialize_context(version="12c", encoding="utf-8")
```

It is recommended to use the Module class as a context manager to release memory allocated during opening of the module.

```
with Module.load("./path/to/form.fmb") as module:
    ... # all actions with form performed here
    module.save() # optionally save changes done to form
```

Rest of the documentation assumes you have a module loaded successfully.

CHAPTER 3

Reading and modifying properties

Once you have a module loaded in Python, you can read/write to object properties defined in the form at will.

```
>>> module.name
'MODULE1'
>>> module.name = module.name[::-1]
>>> module.name
'1ELUDOM'
```

You can read and modify text, numeric and boolean properties. The attribute names in most cases match 1:1 with the ones you can see in the Property Palette when form is opened in Forms Builder.

CHAPTER 4

Iterating over Forms objects

Object hierarchy matches the one you are familiar with in Forms Builder. Canvases, windows, triggers, program units, etc. are also available.

With this information in mind here is how you would iterate over all of the defined data blocks and their respective items.

```
>>> [item.name for data_block in module.data_blocks for item in data_block.items]
['TEXT_ITEM4',
 'TEXT_ITEM284',
 'PUSH_BUTTON285',
 'CHECK_BOX286',
 'RADIO_GROUP288']
```

Creating new objects

Creation of new objects is also supported.

Given a data block, you can create a new item for it. First you must also import the type of object you want to create.

```
>>> from pyoracle_forms import Item
>>> data_block = module.data_blocks[0]
>>> new_item = Item.create(data_block, "NEW_ITEM_NAME")
>>> new_item.name
'NEW_ITEM_NAME'
```

Newly created item will be attached to the data block.

Using older Oracle Forms versions

By default forms API version is assumed to be 12c, and the encoding of .fmb file text as utf-8, to use a different version or encoding you are able to pass other version values to `initialize_context` function Supported versions are 12c, 10g and 6i of the Oracle forms API.

```
>>> initialize_context(version="6i", encoding="cp1257")
```