

Open Recon - Getting started with Python



© Siemens Healthineers AG - All Rights Reserved

Restricted - Unauthorized copying of this file, via any medium is strictly prohibited

(Version Mar 26th 2024)

The [python-ismrmrd-server](#) provides a starting point for development of Python-based Open Recon applications. The [i2i.py](#) and [r2ci.py](#) files in this folder can be directly [used as modules](#) in the server for image-to-image and raw-to-complex-image workflows respectively.

As Open Recon is intended for deployment of a single application per container, the ["config" mechanism in MRD](#) used to select between different modules is not sent from Open Recon. Instead, an [MRD_MESSAGE_CONFIG_FILE](#) message with value [openrecon](#) is always sent for every recon.

When used with the Python server, Open Recon will request a module with name [openrecon](#) (i.e. a module file named [openrecon.py](#)). If this file is not present, then the [default config of invertcontrast](#). The default config can be specified when starting the server [using the --defaultConfig option](#). When using the Python server with Open Recon, the [--defaultConfig](#) option must be specified appropriately or the module must be named [openrecon.py](#). This can be done by setting the [CMD](#) argument [in the Dockerfile](#), e.g

```
CMD [ "python3", "/opt/code/python-ismrmrd-server/main.py", "-v", "-H=0.0.0.0", "-p=9002", "-l=/tmp/python-ismrmrd-server.log", "--defaultConfig=invertcontrast"]
```

It may be desirable to select between multiple different configs/modules when using Open Recon. This is possible by specifying parameters in the Open Recon UI named [config](#) or [customconfig](#). For example, in the parameters section of the JSON configuration:

```
"parameters": [
  {
    "id": "config",
    "type": "choice",
    "label": { "en": "config" },
    "values": [
      {
        "id": "invertcontrast",
        "name": { "en": "invertcontrast" }
      },
      {
        "id": "rgb",
        "name": { "en": "rgb" }
      },
      {
        "id": "simplefft",
        "name": { "en": "simplefft" }
      }
    ]
  }
]
```

```

    ],
    "default": "invertcontrast",
    "information": { "en": "Define the config to be executed by MRD server" }
  },
  {
    "id": "customconfig",
    "label": { "en": "Custom Config" },
    "type": "string",
    "information": { "en": "Custom config file not listed in drop-down menu" },
    "default": ""
  },
},

```

This is [interpreted by the server](#) and overrides the [openrecon](#) config sent by default, with [customconfig](#) taking priority over [config](#) if present.

Getting started with an image-to-image application

1. Clone (or download) the python-ismrmrd-server repository.

```
git clone https://github.com/kspaceKelvin/python-ismrmrd-server.git
```

2. Copy the [i2i.py](#) file from this repo into the [python-ismrmrd-server](#) folder.
3. Change the default config [in the CMD line](#) of the [Dockerfile](#) to indicate [i2i.py](#) should be run unless otherwise overridden.

```
CMD [ "python3", "/opt/code/python-ismrmrd-server/main.py", "-v", "-H=0.0.0.0", "-p=9002", "-l=/tmp/python-ismrmrd-server.log", "--defaultConfig=i2i"]
```

4. Open a command prompt in the folder containing [python-ismrmrd-server](#) (i.e. one folder up) and build the Docker image.

```
docker build --no-cache -t openrecon-i2i -f python-ismrmrd-server/docker/Dockerfile ./
```

5. Make any other desired UI changes to [i2i_json_ui.json](#).
6. Use the [CreateORDockerImage.ipynb](#) notebook to package the OpenRecon app. Update the [jsonFilePath](#) and [baseDockerImage](#) variables with the name of the Docker image created above and the [i2i](#) UI json file.

```
jsonFilePath    = 'i2i_json_ui.json'
baseDockerImage = 'openrecon-i2i'
```

7. Create a documentation PDF file and update the `docsFile` parameter in `CreateORDockerImage.ipynb`.

```
docsFile = 'docs.pdf'
```

8. Run `CreateORDockerImage.ipynb` to build the packaged Open Recon app .zip file. Copy the OpenRecon .zip file into the folder `C:\Program Files\Siemens\Numaris\OperationalManagement\FileTransfer\incoming`, referring to the "Deployment on the Scanner" section in the [README](#) for more information.

Getting started with a raw-to-complex-image application

1. Clone (or download) the `python-ismrmrd-server` repository.

```
git clone https://github.com/kspaceKelvin/python-ismrmrd-server.git
```

2. Copy the `r2ci.py` file from this repo into the `python-ismrmrd-server` folder.
3. Change the default config in the `CMD` line of the `Dockerfile` to indicate `r2ci.py` should be run unless otherwise overridden.

```
CMD [ "python3", "/opt/code/python-ismrmrd-server/main.py", "-v", "-H=0.0.0.0", "-p=9002", "-l=/tmp/python-ismrmrd-server.log", "--defaultConfig=r2ci"]
```

4. Open a command prompt in the folder containing `python-ismrmrd-server` (i.e. one folder up) and build the Docker image.

```
docker build --no-cache -t openrecon-r2ci -f python-ismrmrd-server/docker/Dockerfile ./
```

5. Make any other desired UI changes to `r2ci_json_ui.json`.
6. Use the `CreateORDockerImage.ipynb` notebook to package the OpenRecon app. Update the `jsonFilePath` and `baseDockerImage` variables with the name of the Docker image created above and the `r2ci` UI json file.

```
jsonFilePath      = 'r2ci_json_ui.json'  
baseDockerImage  = 'openrecon-r2ci'
```

7. Create a documentation PDF file and update the `docsFile` parameter in `CreateORDockerImage.ipynb`.

```
docsFile = 'docs.pdf'
```

8. Run CreateORDockerImage.ipynb to build the packaged Open Recon app .zip file. Copy the OpenRecon .zip file into the folder `C:\Program Files\Siemens\Numaris\OperationalManagement\FileTransfer\incoming`, referring to the "Deployment on the Scanner" section in the [README](#) for more information.

Disclaimer

Open Recon is to add clinical reconstructions to the system, if signed and released for clinical use by Siemens Healthineers. Any other recon used e.g., by researchers is automatically labelled not for diagnostic use, which may require observation of national regulations.