# Installation

Python connector is a wrapper for the *SAP NetWeaver RFC Library* and you need to obtain and install it first.

If Python is not already installed on your system, you need to download and install Python as well.

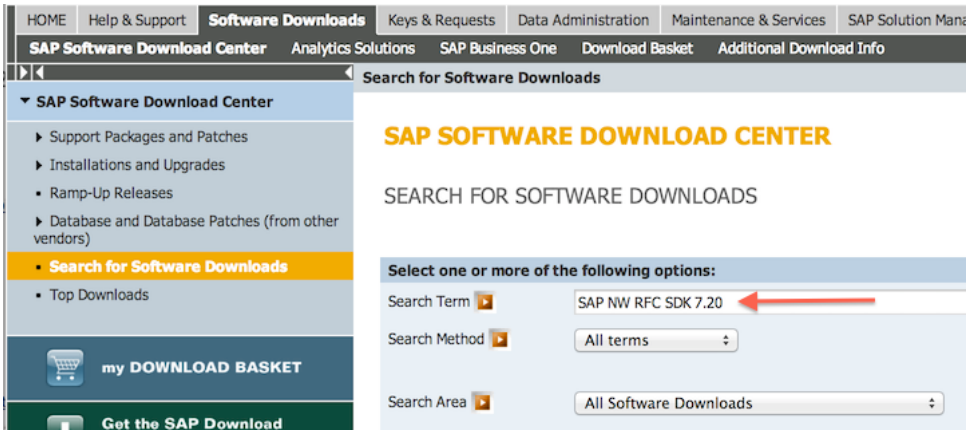
After having *SAP NW RFC Library* and Python installed on your system, you can download and install one of provided `pyrfc` eggs, relevant for your platform and start using `pyrfc`.

You can also clone this repository and build `pyrfc` from the source code, following *Building from source*.

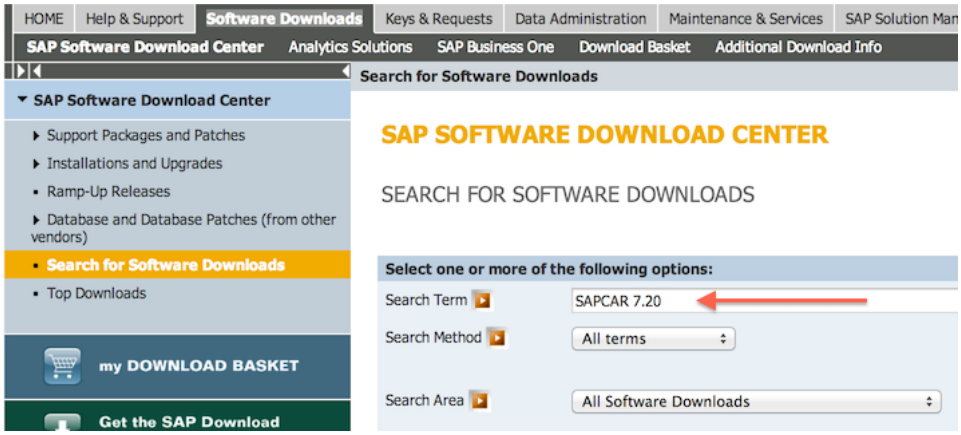
## SAP NW RFC Library Installation

The entry page for *SAP NetWeaver RFC library* is SAP Service Marketplace (SMP), http://service.sap.com/rfc-library, with detailed instructions how to download, use and compile.

Basically, you should search for `SAP NW RFC SDK 7.20`, in `Software Downloads` of SAP Software Download Center on SMP Support Portal, download SAP NW RFC Library adequate for your platform and Python version combination (see the matrix below) and unpack using `SAPCAR` archive utility.




`SAPCAR` can be donwloaded from the SMP as well and you should search for `SAPCAR 7.20`




Which SAP NW RFC Library version is relevant for your platform? Here are platform/Python combinations tested so far:

| Platform | Python version | NetWeaver RFC Library (SMP) | Filename (SMP) | Python egg |
|---|---|---|---|---|
| Windows | Python 2.7 (32bit) | *Windows Server on IA32 32bit* | `NWRFC_20-20004566.SAR` | pyrfc-1.9.3-py2.7-win32.egg |
| Windows | Python 2.7 (64bit) | *Windows on x64 64bit* | `NWRFC_20-20004568.SAR` | pyrfc-1.9.3-py2.7-win-amd64.egg |
| Linux | Python 2.7 (64bit) | *Linux on x86_64 64bit* | `NWRFC_20-20004565.SAR` | pyrfc-1.9.3-py2.7-linux-x86_64.egg |

TABLE OF CONTENTS

- Introduction
- Installation
  - SAP NW RFC Library Installation
  - Python Connector Installation
  - Virtual environments
  - Problems
- Client scenario
- Server scenario
- Security
- Building from source
- Remarks
- Bibliography
- pyrfc
- Change log

> **Note**
>
> - *SAP NW RFC Library* is fully backwards compatible and it is reccomended using the newest version also for older backend system releases
> - SMP search terms and filenames given here will not be regularly updated, you should always search for current version or filename in `Software Downloads`.
> - The server functionality is currently not working under Windows 32bit

The Python connector relies on *SAP NW RFC Library* and must be able to find library files at runtime. Therefore, you might either install the *SAP NW RFC Library* in the standard library paths of your system or install it in any location and tell the Python connector where to look.

Here are configuration examples for Windows and Linux operating systems.

### Windows

1. Create an directory, e.g. `c:\nwrfcsdk`.
2. Unpack the SAR archive to it, e.g. `c:\nwrfcsdk\lib` shall exist.
3. Include the `lib` directory to the library search path on Windows, i.e. *extend* the `PATH` environment variable.

### Linux

1. Create the directory, e.g. `/usr/sap/nwrfcsdk`.
2. Unpack the SAR archive to it, e.g. `/usr/sap/nwrfcsdk/lib` shall exist.
3. Include the `lib` directory in the library search path:

   - As `root`, create a file `/etc/ld.so.conf.d/nwrfcsdk.conf` and enter the following values:

     ```
     # include nwrfcsdk
     /usr/sap/nwrfcsdk/lib
     ```

   - As `root`, run the command `ldconfig`.

## Python Connector Installation

### Windows

- If not already installed, you need to install Python first.

  First, decide whether you want to go with the 32bit or 64bit version and use standard Windows installers

  Python 2.7 (32 bit), http://www.python.org/ftp/python/2.7.6/python-2.7.6.msi

  Python 2.7 (64 bit) http://www.python.org/ftp/python/2.7.6/python-2.7.6.amd64.msi

  Add Python and Scripts directories to `PATH` environment variable, e.g.

  ```
  set PATH=c:\Python27;c:\Python27\Scripts;%PATH%
  ```

- Install `easy_install`

  Use the `distribute` implementation of `easy_install` by downloading https://bootstrap.pypa.io/ez_setup.py and running

  ```
  python ez_setup.py
  ```

  > **Note**
  >
  > At this point you may like to install the pip package which extends the functionality of `easy_install`. However, `pip` cannot handle binary build distributions, which will be used later.
  >
  > If you are in a internal network that uses a proxy to access resources from the internet, you may encounter *connection problems*.

- Virtual environment (optional)

  You may now create an *virtual environment* and activate it.

- Install the Python connector

  Open the command prompt with administrator rights, change to the `pyrfc\dist` directory and install adequate `pyrfc` egg. You need administrator rights, otherwise `easy_install` will open a new window and close it after execution – leaving you without the option to see what was done or what was the error.

```
easy_install <egg name>
```

  Please look up the correct *egg name* depending on your platform and Python version.

- Run `python` and type `from pyrfc import *`. If this finishes silently, without oputput, the installation was successful.

### Python on Linux

- Install Python 2.7 (64bit, usually the default) via your preferred package manager
- Install `easy_install`

  Use the `distribute` implementation of `easy_install` by downloading https://bootstrap.pypa.io/ez_setup.py and running

```
python ez_setup.py
```

> **Note**
>
> At this point you may like to install the pip package which extends the functionality of `easy_install`. However, `pip` cannot handle binary build distributions, which will be used later.
>
> If you are in a internal network that uses a proxy to access resources from the internet, you may encounter *connection problems*.

- Virtual environment (optional)

  You may now create an *virtual environment* and activate it.

- Install the Python connector:

```
easy_install <egg name>
```

  Please look up the correct *egg name* depending on your platform and Python version.

- Run `python`, type `from pyrfc import *` and it it finishes silently, without any output, the installation was successful.

## Virtual environments

We recommend using a virtual environment for the installation. This allows you to isolate the Python connector installation from your system wide Python installation.

We will now show the example usage for a Windows user that wants to create a virtual environment in `C:\PythonVE\py27-pyrfc`.

1. Install `virtualenv` on your system.

```
C:\>pip virtualenv
```

2. Open a command prompt and change to a directory where you want to create a virtual environment and create a virtual environment.

```
C:\>cd PythonVE
C:\PythonVE\>virtualenv --distribute --no-site-packages py27-sapwnrfc2
```

   (Since `virtualenv` version 1.7, the `--no-site-packages` option is the default and can be omitted.)

3. Activate the environment via

```
C:\PythonVE\>cd py27-pyrfc
C:\PythonVE\py27-pyrfc\>Scripts\activate.bat
(py27-pyrfc) C:\PythonVE\py27-pyrfc\>
```

(On Linux use `source bin/activate`.)

4. After working on your project, you leave the virtual environment with

```
(py27-pyrfc) C:\PythonVE\py27-pyrfc\>deactivate
C:\PythonVE\py27-pyrfc\>
```

# Problems

## Behind a Proxy

If you are within an internal network that accesses the internet through an HTTP(S) proxy, some of the shell commands will fail with urlopen errors, etc.

Assuming that your HTTP(S) proxy could be accessed via `http://proxy:8080`, on Windows you can communicate this proxy to your shell via:

```
SET HTTP_PROXY=http://proxy:8080
SET HTTPS_PROXY=http://proxy:8080
```

or permanently set environment variables.

## SAP NW RFC Library Installation

1. `ImportError: DLL load failed: The specified module could not be found.`

   (Windows) This error indicates that the Python connector was not able to find the C connector on your system. Please check, if the `lib` directory of the C connector is in your `PATH` environment variable.

2. `ImportError: DLL load failed: %1 is not a valid Win32 application.`

   (Windows) This error occurs when SAP NW RFC Library 64bit version is installed on a system with 32bit version Python.

## Environment variables

### Windows

The environment variable may be set within a command prompt via the `set` command, e.g.

- `set PATH=%PATH%;C:\nwrfcsdk\lib` (extend PATH with the C connector lib)
- `set HTTPS_PROXY=proxy:8080` (setting an proxy for HTTPS communication)

When the command prompt is closed, the environment variable is reset. To achieve a persistent change of the environment variable, do the following (Windows 7):

1. Open the Start Menu and type `environment` into the search box.
2. A window opens in which the user variables are displayed in the upper part and the system variables in the lower part. You may select and edit the desired variable.
3. The modified variables are used when a *new* command prompt is opened.