<http://sap.github.io/PyRFC/install.html>

[pyrfc 2.0.2 documentation](http://sap.github.io/PyRFC/index.html)

[PREVIOUS](http://sap.github.io/PyRFC/intro.html) | [NEXT](http://sap.github.io/PyRFC/client.html) | [INDEX](http://sap.github.io/PyRFC/genindex.html)

# Installation

If [SAP NetWeaver RFC SDK](https://support.sap.com/en/product/connectors/nwrfcsdk.html) and Python are already installed on your system, you can pip install the pyrfc wheel from the [latest release](https://github.com/SAP/PyRFC/releases/latest), or clone this repository and build pyrfc from the source code, following [Building from source](http://sap.github.io/PyRFC/build.html#build).

Use the Python 3 and the latest pyrfc and SAP NW RFC SDK release (fully backwards compatible).

## SAP NW RFC SDK Installation

If SAP NW RFC SDK is already installed on your system, you may verify the installation by running the rfcexec utility, without any parameter.

The error message like below indicates that SAP NW RFC SDK installation is technically correct, expecting more input parameters. Different error message may be caused by missing Windows C++ binary for example, or another installation inconsistency:

$ cd /usr/local/sap/nwrfcsdk/bin

$ ./rfcexec

Error: Not all mandatory parameters specified

Please start the program in the following way:

rfcexec -t -a <program ID> -g <gateway host> -x <gateway service>

-f <file with list of allowed commands> -s <allowed Sys ID>

The options "-t" (trace), "-f" and "-s" are optional.

Information on where to download the SAP NW RFC SDK you may find [here](https://support.sap.com/en/product/connectors/nwrfcsdk.html) .

The PyRFC connector relies on SAP NW RFC SDK and must be able to find the library files at runtime. Therefore, you might either install the SAP NW RFC SDK 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, Linux and macOS operating systems.

### Windows

1. Create the SAP NW RFC SDK home directory, e.g. c:\nwrfcsdk
2. Set the SAPNWRFC\_HOME env variable: SAPNWRFC\_HOME=c:\nwrfcsdk
3. Unpack the SAP NW RFC SDK archive to it, e.g. c:\nwrfcsdk\lib shall exist.
4. Include the lib directory to the library search path on Windows, i.e. [extend](http://sap.github.io/PyRFC/install.html#install-problems-envvar-win) the PATH environment variable.

Add c:\nwrfcsdk\lib to PATH.

### Linux

1. Create the SAP NW RFC SDK home directory, e.g. /usr/local/sap/nwrfcsdk
2. Set the SAPNWRFC\_HOME env variable: SAPNWRFC\_HOME=/usr/local/sap/nwrfcsdk
3. Unpack the SAP NW RFC SDK archive to it, e.g. /usr/local/sap/nwrfcsdk/lib shall exist.
4. 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/local/sap/nwrfcsdk/lib
   * As root, run the command ldconfig. To check if libraries are installed:
   * $ ldconfig -p | grep sap *# should show something like:*
   * libsapucum.so (libc6,x86-64) => /usr/local/sap/nwrfcsdk/lib/libsapucum.so
   * libsapnwrfc.so (libc6,x86-64) => /usr/local/sap/nwrfcsdk/lib/libsapnwrfc.so
   * libicuuc.so.50 (libc6,x86-64) => /usr/local/sap/nwrfcsdk/lib/libicuuc.so.50
   * libicui18n.so.50 (libc6,x86-64) => /usr/local/sap/nwrfcsdk/lib/libicui18n.so.50
   * libicudecnumber.so (libc6,x86-64) => /usr/local/sap/nwrfcsdk/lib/libicudecnumber.so
   * libicudata.so.50 (libc6,x86-64) => /usr/local/sap/nwrfcsdk/lib/libicudata.so.50
   * libgssapi\_krb5.so.2 (libc6,x86-64) => /usr/lib/x86\_64-linux-gnu/libgssapi\_krb5.so.2
   * libgssapi.so.3 (libc6,x86-64) => /usr/lib/x86\_64-linux-gnu/libgssapi.so.3
   * $

### macOS

The macOS firewall stealth mode is by default active, blocking the ICMP protocol based network access to Macbook. Applications like Ping do not work by default ([Can’t ping a machine - why?](https://discussions.apple.com/thread/2554739)) and the stealth mode must be disabled:

sudo /usr/libexec/ApplicationFirewall/socketfilterfw --setstealthmode off

1. Create the SAP NW RFC SDK root directory /usr/local/sap/nwrfcsdk (this location is fixed, more info below)
2. Set SAPNWRFC\_HOME environment variable to that location: SAPNWRFC\_HOME=/usr/local/sap/nwrfcsdk
3. Unpack the SAP NW RFC SDK archive to it, e.g. /usr/local/sap/nwrfcsdk/lib shall exist.
4. Set the remote paths in SAP NW RFC SDK by running [paths\_fix.sh](https://github.com/SAP/PyRFC/blob/master/ci/utils/paths_fix.sh) script.
5. Unzip the unchar.zip file attached to [SAP OSS Note 2573953](https://launchpad.support.sap.com/#/notes/2573953) and copy the uchar.h file to SAP NW RFC SDK include directory

This location is fixed to the default /usr/local/sap/nwrfcsdk/lib rpath, embedded into node-rfc package published on npm.

After moving SAP NW RFC SDK to another location on your system, the rpaths must be adjusted in SAP NW RFC SDK and in pyrfc.so libraries.

For SAP NW RFC SDK, set the SAPNWRFC\_HOME env variable to new SAP NW RFC SDK root directory and re-run the above script.

For pyrfc:

$ unzip unzip pyrfc-2.0.1-cp38-cp38-macosx\_10\_15\_x86\_64.whl

$ cd pyrfc

$ install\_name\_tool -rpath /usr/local/sap/nwrfcsdk/lib <new path> \_pyrfc.cpython-38-darwin.so

## Python Connector Installation

Download the wheel from your platform, from the [latest release](https://github.com/SAP/PyRFC/releases/latest) and pip install.

Using virtual environments you can isolate Python/PyRFC projects, working without administrator privileges.

### Windows

* If not already installed, install the Python first: <https://www.python.org/downloads/windows/>

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

set PATH=c:\Python37;c:\Python37\Scripts;%PATH%

* Install pip if not already included: <https://pip.pypa.io/en/stable/installing/>
* Install the Python connector from the [latest release](https://github.com/SAP/PyRFC/releases/latest)
* wget https://github.com/SAP/PyRFC/releases/download/2.0.0/pyrfc-2.0.0-cp38-cp38-win\_amd64.whl
* pip install pyrfc-1.9.97-cp37-cp37m-macosx\_10\_14\_x86\_64.whl

Please look up the correct wheel 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.

### Linux

* Install Python 3
* Install pip if not already included: <https://pip.pypa.io/en/stable/installing/>
* Install the Python connector from the [latest release](https://github.com/SAP/PyRFC/releases/latest)
* wget https://github.com/SAP/PyRFC/releases/download/2.0.0/pyrfc-2.0.0-cp38-cp38-linux\_x86\_64.whl
* pip install pyrfc-1.9.94-cp37-cp37m-linux\_x86\_64.whl

Please look up the correct wheel 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.

### macOS

The macOS system version of Python is usually the older one and using wirtual environments, like [pyenv](https://github.com/pyenv/pyenv) for example, is recommended:

pyenv install 3.8.0

pyenv virtualenv 3.8.0 py380

Install the Python connector the same way like for Linux.

## 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.

1. 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

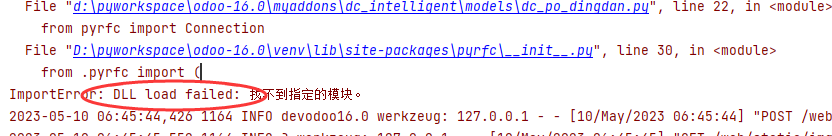
The environment variable may be set within a command prompt via the set command, e.g.

* 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.

Q1:在安装PYRFC后，执行程序时，出现如下提示。



A1:安装 vcredist2013\_x64