z/OS Miniconda Documentation

Miniconda and conda are open source products. Conda is a package manager and Miniconda is an installer for conda. z/OS Miniconda is a version of Miniconda that contains z/OS-specific ports of conda, Bash, Python, and a few other packages required to run conda. This document describes the z/OS-specific aspects of z/OS Miniconda and z/OS conda and provides the minimum instructions needed to effectively get started with Miniconda and conda on z/OS. For full conda and Miniconda documentation refer to the conda project user guide.

z/OS Specifics

There are some key characteristics of the z/OS environment to be aware of:

- All z/OS Miniconda components are 64-bit ASCII applications, except z/OS Perl, which was built in EBCDIC mode.
- Since z/OS Miniconda is an ASCII environment running on a platform with a default EBCDIC code page, file tagging is used extensively. Be aware that newly created files in this environment must be properly tagged for everything to run smoothly.
- z/OS Miniconda commands should be launched in z/OS Bash, which is included in the z/OS Miniconda installation.

System Requirements

Ensure that the following software and hardware requirements for z/OS Miniconda have been met:

- IBM z/OS V2.2 or later
- 2827-xxx (IBM zEnterprise EC12) and 2828-xxx (IBM zEnterprise BC12) models or later
- Free space: 610 MB available for Miniconda installation alone; 2.6 GB available for Miniconda plus the Rocket Open AppDev for z/OS open source solution bundle.

Installing Miniconda from Rocket Community Portal

Downloading z/OS Miniconda

To download z/OS Miniconda from the Rocket Community Portal, complete the following steps:

1. Sign in to the Rocket Community Portal.
   https://my.rocketsoftware.com/RocketCommunity
2. Click **Downloads** from the navigation menu. A list of available Rocket products for your account appears.
3. Select **z/OpenSource**.
4. Type 'Miniconda' in the search box to filter the list of available downloads.
5. Click the filtered **ID** number. A dialog window appears with the files that are required for setup.
6. Click each file to download them to your system.
7. Using a FTP or SFTP client transfer the setup files to z/OS using **binary** mode:
    a. miniconda-zos-<X.Y>-<yyyy-mm-dd>.run
    b. appdev_manifest-<X.Y>.txt

Installing z/OS Miniconda

In multi-user environments end users shouldn't have write access to the Miniconda installation directory. It is recommended to create a z/OS user which will install and configure Miniconda. This Miniconda administrator doesn't require any special permission nor superuser access.

To install Miniconda on z/OS, complete the following steps:

1. Login to your z/OS system using ssh client (e.g. PuTTy) and change directory to the path where you saved Miniconda installation files.

2. Set tags for these files:
   chmod +x miniconda-zos-<X.Y>-<yyyy-mm-dd>.run
   chtag -tc 819 appdev_manifest-<X.Y>.txt
3. Set following environment variables to be able to run the install script:
   export _BPXK_AUTOCVT=ON
   export _CEE_RUNOPTS='FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)'
4. Execute the self-extracting installer at the prompt:
   ./miniconda-zos-<X.Y>-<yyyy-mm-dd>.run --path <miniconda_installation_path>
5. Confirm installation directory and conda channel server configuration in the interactive installation

Note: Optional path argument specifies installation directory. If you don't specify a path Miniconda will be installed into your home directory in a folder named "miniconda".

Each end user who needs access to Miniconda should have following settings in a Bash startup file:

```
export _BPXK_AUTOCVT=ON
export _CEE_RUNOPTS="FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"
. <miniconda_installation_path>/etc/profile.d/conda.sh
```

Examples:

1. Login shell specified in RACF is /bin/sh (default case). Then use next configuration:
   a. Add the following lines to `$HOME/.profile`:
      ```
      export _BPXK_AUTOCVT=ON
      export _CEE_RUNOPTS="FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"
      SHELL=<miniconda_installation_path>/bin/bash
      exec $SHELL
      ```
   b. Add the following line to `$HOME/.bashrc`:
      ```
      . <miniconda_installation_path>/etc/profile.d/conda.sh
      ```
2. RACF login shell is Bash. Then add the following lines to `$HOME/.profile`:
   ```
   export _BPXK_AUTOCVT=ON
   export _CEE_RUNOPTS="FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"
   . <miniconda_installation_path>/etc/profile.d/conda.sh
   ```

When you restart ssh-session, Miniconda will be ready to use.

Using conda to download and install z/OS products

Once z/OS Miniconda is installed on your system you can issue conda commands to create environments and download and install z/OS products. This section describes the basic concepts and commands to do this and introduces two product repositories (called channels in the conda world) Rocket has set up and from which you can obtain z/OS products. It is important to note again that only a basic introductory commands are documented herein. For full conda documentation, refer to the conda project user guide.

Note: Although conda can be used to install products such as Git as well as language packages such as scikit-learn for Python, in conda terminology, what you download and install from a channel is always referred to as a *package*.

Rocket's conda channels for z/OS products

Conda channels are the locations (identified as URLs) where products and packages are stored. They serve as the base for hosting and managing products and packages you can install. Rocket has set up two internet channels specifically to host open source products that have been modified as needed and tested and validated to function on z/OS. Generally the same products can be found in both channels however Rocket's secure channel is updated more frequently with the latest version updates and security patches. If you installed Miniconda with default parameters, your installation will have all required settings to connect to Rocket's secure conda server.

You can browse what is available in any channel using a web browser however you cannot install via web browser.

Rocket public conda channel for z/OS Open source

The channel https://anaconda.org/zoss-appdev has been set up and is maintained by Rocket and is open and available to anyone.

Rocket secure conda channel for z/OS Open source

The channel https://condaserver.rocketsoftware.com/ is a private channel hosted by a secure authenticated server on premise at Rocket and available to Rocket customers with support contracts.

To obtain an account on Rocket's secure conda channel server, please submit a request from our open source product page.

Creating a conda environment

A **conda environment** is a directory that contains a specific collection of **conda** products and packages that you have installed. For example, you may have one **environment** for testing and another **environment** for production. When you install a product or package using conda, you install it into an environment.

Miniconda is initially installed with a base environment containing a few packages needed for its operation. However it is not recommended to install programs in the base environment. Therefore, it is recommended that you create separate environments to keep programs isolated from each other.

To create a new environment, complete the following steps:

1. Enter the following command to create the environment:

   ```
   ~ conda create -n <new_environment_name>
   ```

Note: The new environment is now created, but it is not yet available for use until you activate it. You can activate an environment by entering the command:

```
~ conda activate <new_environment_name>
```

If you want to use only base environment just type "conda activate" or "conda activate base"

For more information about environments, review the conda project documentation at: https://docs.conda.io/projects/conda/en/latest/user-guide/getting-started.html#managing-environments

Installing products or packages with conda

Full details described in the conda project documentation at: https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-pkgs.html#installing-packages

It is possible to install pure python packages from any anaconda channel, however we recommend installing only packages which were ported and tested on z/OS.

Before installing a product or package be sure you have an activated environment (see Creating a conda environment).

To install a product or package to the current active environment, complete the following steps:

1. (Skip this step if you are using public channel or you are using Rocket private channel and have already logged in during the installation process)
   Login to Rocket's Anaconda repository:

   ```
   ~ conda repo login
   ```

   Enter user and password which was provided by Rocket Sales. Once logged in you don't need to do it again.

2. Enter the following command:

   ```
   ~ conda install <product or package name>
   ```

   If you want to install all available packages and tools, the command is:

   ```
   ~ conda install --file appdev_manifest-<X.Y>.txt
   ```

   If you install packages from the public channel, the command is:

   ```
   ~ conda install -c https://conda.anaconda.org/zoss-appdev <product or package name>
   ```

   Review the list of packages to be installed and follow the on-screen prompts.

Note: You do not need to worry about dependencies. Conda does that automatically. For example, Git has several dependencies but you only need to install Git and its dependencies will also be automatically installed.

Note: To browse what is available on an internet channel you can visit the URL with a web browser

Note: To search a channel for products or packages by name use the conda search command. For example those starting with letter 'g':

```
~ conda search g*
```

`Note:` You can combine environment creation and product installation in one command. For example to install git and its dependencies into a new environment named 'test_env':

```
~ conda create -n test_env git
```

Updating z/OS Miniconda

When a new version of z/OS Miniconda is available on the Rocket Community Portal, you can update your version by running the new setup script. Previously existing environments are not effected by the update. The update only makes changes to the packages in the base environment.

To update your version, complete the following steps:

1. Download the files from the Rocket Community Portal.

2. Transfer the files to z/OS via FTP or SFTP protocol using **binary** mode.
3. Specify the path to the setup script and run the following command:

```
./miniconda-zos-<X.Y>-<yyyy-mm-dd>.run --path <miniconda_installation_path>
```

Note: If you previously used a path argument to specify an installation directory, use the same path argument value for the update. If you did not specify a path, the update will be installed in your home directory in the folder named 'miniconda'.

Migrating to z/OS Miniconda

If you are familiar with Rocket z/OS ported languages and tools, you might have some of them already installed. Other than z/OS conda, via the z/OS Miniconda download, Rocket no longer support z/OS ports delivered via the Rocket Support Portal. To migrate installed z/OS ports that were installed prior to the introduction of z/OS Miniconda you need to:

1. Remove paths to previously installed tools from PATH, LIBPATH, MANPATH environment variables.
2. Install Miniconda according to the instructions above.
3. Make sure that Miniconda installation is working on your system.
4. Delete folder with previously installed tools.

Deploying a custom channel for Rocket Open AppDev for Z

If you want developers in your organization to be able to download and install Rocket's z/OS open source ports without going outside your network (e.g. if your mainframe systems are air-gapped) you will need to deploy an on-premise custom conda channel. The instructions below will set up a custom channel and populate it with the contents of Rocket Open AppDev for Z solution bundle.

Note: This option is only available to customers on a support contract for Rocket Open AppDev for Z.

Downloading the channel installer

To download the channel installer, complete the following steps:

1. Sign in to the Rocket Community Portal.
   https://my.rocketsoftware.com/RocketCommunity
2. Click **Downloads** from the navigation menu. A list of available Rocket products for your account appears.
3. Select **z/OpenSource.**
4. Type 'Rocket Open AppDev for Z' in the search box to filter the list of available downloads.
5. Click the filtered **ID** number. A dialog window appears with the files that are required for the setup of the custom channel.
6. Choose **appdev_local-<X.Y>-<yyyy-mm-dd>.run** for download (this assumes you have already installed z/OS Miniconda).
7. Use an FTP or SFTP client to transfer downloaded files to z/OS using **binary** mode.

Running the channel installer

To run the installer complete the following steps:

1. Login to your z/OS system using ssh client (e.g. PuTTy) and change directory to the path where you saved the installer (appdev_local-<X.Y>-<yyyy-mm-dd>.run).
2. Set execute bit and file tag for the installer:

```
chmod +x appdev_local-<X.Y>-<yyyy-mm-dd>.run
chtag -b appdev_local-<X.Y>-<yyyy-mm-dd>.run
```

3. Activate the conda base environment and run the self-extracting installer at the prompt:

```
conda activate
./appdev_local-<X.Y>-<yyyy-mm-dd>.run --path <path_to_local_appdev_channel>
```

4. Respond to prompts to confirm the installation directory and .condarc update during the interactive installation session

Note: The optional path argument is used to specify the installation directory. If not specified the channel will be installed to your home directory in a folder named "miniconda/channels".

Using your custom channel

If you selected "yes" during the channel installation when asked to update .condarc, as a developer, you can now use conda commands without specifying the channel path, e.g.:

- `conda search <package>`
- `conda install <package>`
- `conda create -n <env_name> <package>`
- ...

As a convenience, developers wanting an environment with the full contents of the Rocket Open AppDev for Z solution bundle can use the optional file argument during environment creation:

- `conda create -n appdev_bundle --file appdev_manifest-<X.Y>.txt`

If you selected "no" during channel installation, you will need to specify your custom channel path in each conda install/create command, e.g.:

- `conda search -c file:///<path_to_local_appdev_channel> <package>`
- `conda install -c file:///<path_to_local_appdev_channel> <package>`
- `conda create -n appdev_bundle --file appdev_manifest-<X.Y>.txt -c file:///<path_to_local_appdev_channel>`
- …