# Configuration

Before the Data Vault Pipeline Description Compiler *dvpdc* can be used, simple configurations must be made. The goal is that the compiler can be used from any directory without us having to navigate into the project itself first.

The compiler is a Python script that is controlled by a batch file. The batch file is called “dvpdc.bat” and before we can run it from any directory, the directory must be included in the environment variables.

To do this, we type “edit system environment variables” in the search field and open the item of the same name. (see Figure 1) We then open the environment variables and select and edit the system variable “Path” in the lower area.

* In the next point “Edit environment variable” we create an environment variable with “New”. Here we insert the path where the batch file dvpdc.bat is located. An example might look like this: C:\data\_vault\_pipelinedescription\processes\dvpdc

From now on you can use the data vault pipeline description compiler in any directory. Just open the command prompt *cmd* and type in dvpdc. But before you can use the compiler you need to add some parameters.

Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Figure 1 System Environment Variables

A screenshot of a computer program

Description automatically generated

Figure 2 System Properties

Ein Bild, das Text, Software, Webseite, Website enthält.

Automatisch generierte Beschreibung

Figure 3 Edit system variables

# Parameter

Executing the batch file only works in combination with certain parameters. If you want to call the batch file you need to include a dvpd file. This dvpd file is in a json format and the call should look like this ***dvpdc dvpd\_filename***. Additionally you can add more parameters. The distinction is made between input and output paramters.

The following parameters are available for selection:

|  |  |  |
| --- | --- | --- |
| **argument** | **description** | **example** |
| ***input*** |  |  |
| --ini\_file | your ini file .ini | --ini\_file= “C:\data\_vault\_pipelinedescription\dvpdc.ini“ |
| --model\_profile\_directory | Path to your model profiles | --model\_profile\_directory= “C:\dvpd\model“ |
| ***output*** |  |  |
| --dvpi\_directory | Path where your dvpi is exported | *Files created:* dvpi.json |
| --log\_directory | Path where your logs are exported | *Files created*: dvpdc.log & dvpisum.txt |
| ***debug*** |  |  |
| --print\_brain | When set, the compiler will print its internal data structure to stdout |  |
|  |  |  |

Your ini file should cover all arguments so you only need to maintain them in one place. Nevertheless you can always overwrite your ini file arguments by including the argument after your script call. So your dvpi directory will be the one you included after your –dvpi\_directory call, not the one you have in your ini file.

# Ini File