

## Problem

Why can't I import mod1 and mod2 from the same package when they aren't in the same directory?

## Solution

The Python import statement searches the PYTHONPATH for modules. If the module is a Python script file in the *site-packages* directory, Python will find it wherever it first appears in the PYTHONPATH.

For nested modules, those of the form `Level1.Level2.Level3`, the behaviour is a bit different. Nested modules, such as these, are implemented as directories within the PYTHONPATH. So `Level1.Level2.Level3` maps to `<some-dir-in-the-PYTHONPATH>/Level1/Level2/Level3.py`. In order to load this module, the Python interpreter must first load `Level1` and `Level1.Level2`. Once the upper level modules are loaded, the interpreter looks in the `__path__` of the upper level module for all lower level modules. In other words, the interpreter will only look in the **first** place it found the highest level module for the lower level modules.

As of ACS 6.0.3, there is an import function that will search the entire PYTHONPATH for modules. To use it, simply add:

```
import AcsutilPy.ACSImport
```

as your first import in your Python code and all subsequent imports will search all directories.

**NOTE:** If the same module appears twice in the PYTHONPATH, the first one found will be loaded.

-- [ArneGrimstrup](#) - 21 Feb 2007

The new import function has been enabled in the Python container as of ACS 6.0.3. If you are using a Python component and need the searching behaviour, there is no need to modify your code.

-- [ArneGrimstrup](#) - 29 May 2007

The Python library module pkgutil provides a much more reliable method to extend the search path. Simply add:

```
from pkgutil import extend_path
__path__ = extend_path(__path__, __name__)
```

To your module's `__init__.py` file and all instances of that module's name in the PYTHONPATH will be included in the search. They should all be at the same level in the directory tree. For example, if your PYTHONPATH is `top1:top2:top3` and you run the extension code for a module "mod" in `top1`, your module's search path could have `top1/mod`, `top2/mod` and `top3/mod` but not `top2/inter/mod`.

## Related articles

- [How can more people do development with ACS on the same machine without disturbing each other?](#)
- [Which ports are used by ACS?](#)
- [Problems connecting to ACS servers on a remote machine: bad /etc/hosts](#)
- [Why does the getComponent method of ZLegacy/ACS.ContainerServices return an object of type None?](#)
- [Why are some of my print statements not showing up in the container output section of acscommandcenter?](#)