Article
Jimmy Xu · Oct 11, 2022 2m read

ZPM Simple Implementation Cookbook

ZPM is designed to work with applications and modules for InterSystems IRIS Data Platform. It consists of two components, the ZPN Client which is a CLI to manage modules, and The Registry which is a database of modules and meta-information. We can use ZPM to search, install, upgrade, remove and publish modules. With ZPM you can install ObjectScript classes, Frontend applications, Interoperability productions, IRIS BI solutions, IRIS Datasets or any files such as Embedded Python wheels.

Today this cookbook will go through 3 sections:

1. Install ZPM
2. Generate Module
3. Find, Install, publish modules inside the Registry

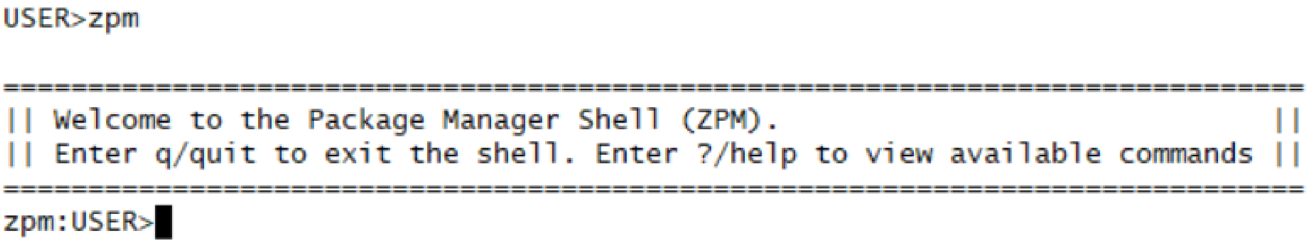
1. Install ZPM

- Download latest version of ZPM (this should be one single XML file) Download link
- Import XML you have downloaded to IRIS and it can only be deployed into IRIS open IRIS terminal and enter

write $SYSTEM.OBJ.Load("C:\zpm.xml", "c")

Note "C:\zpm.xml" is the path of the XML file downloaded, this step could take a while.

- After finished installing, simply type *zpm*, press enter, you will see you are in zpm shell



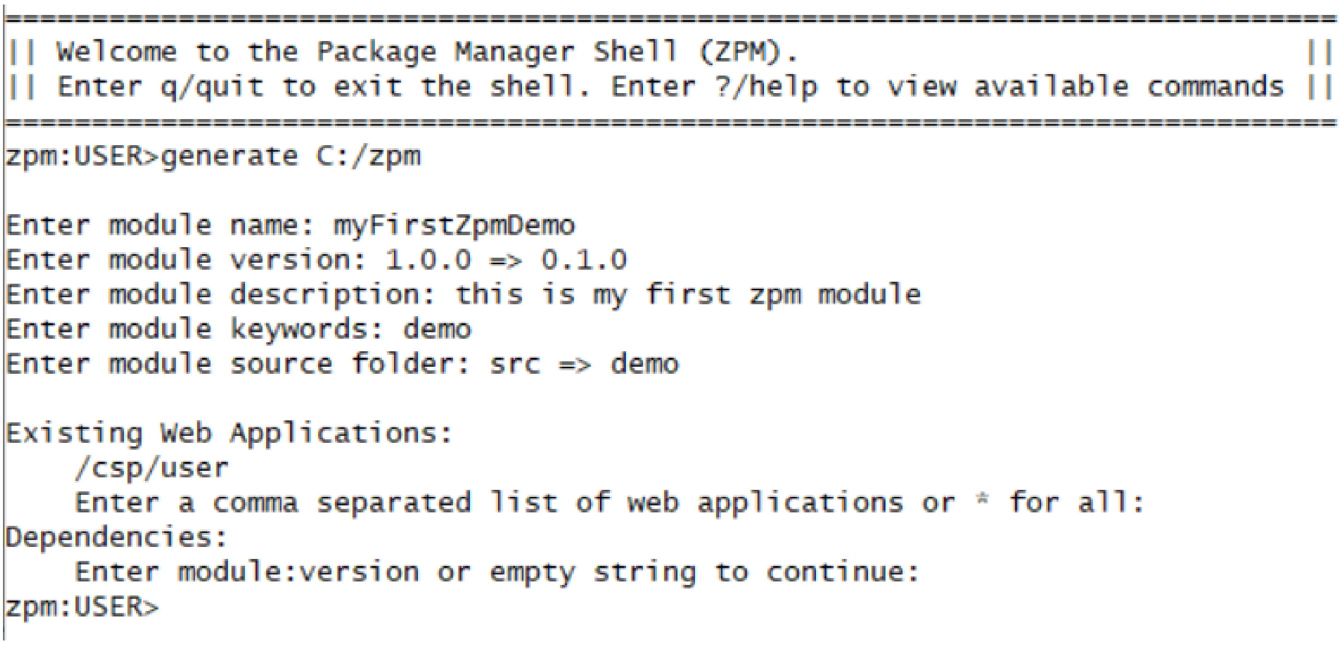
```
USER>zpm

=============================================================================
|| Welcome to the Package Manager Shell (ZPM).                             ||
|| Enter q/quit to exit the shell. Enter ?/help to view available commands ||
=============================================================================
zpm:USER>
```

2. Generate Module

Before we start generating module, we need to prepare a folder that has one or more files ready to load, Therefore I have created a folder under C drive called zpm.

Execute the command *generate C:/zpm*

After you specific all the necessary, your first module has been generated successfully, also you will see

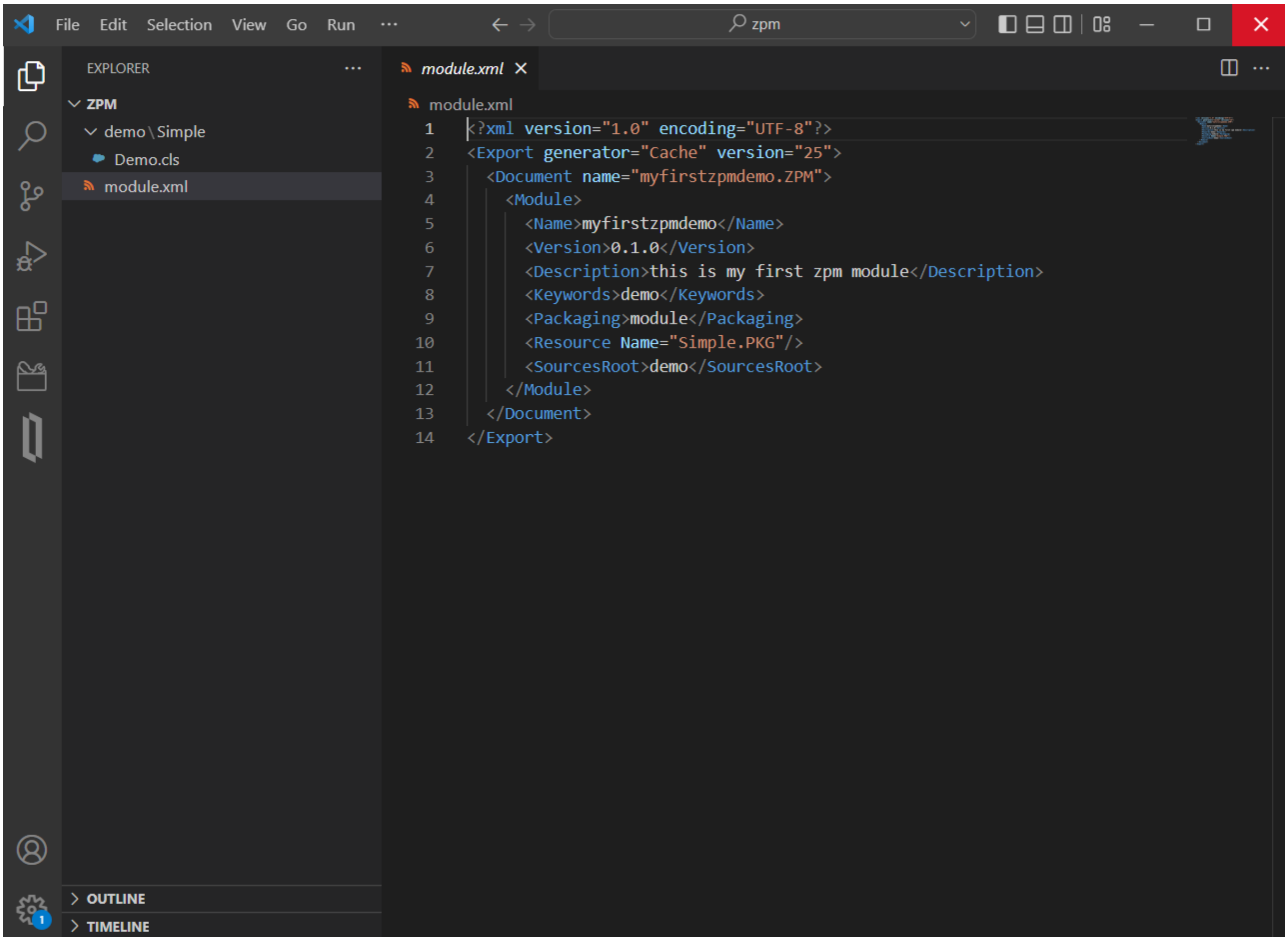


```
=============================================================================
|| Welcome to the Package Manager Shell (ZPM).                             ||
|| Enter q/quit to exit the shell. Enter ?/help to view available commands ||
=============================================================================
zpm:USER>generate C:/zpm

Enter module name: myFirstZpmDemo
Enter module version: 1.0.0 => 0.1.0
Enter module description: this is my first zpm module
Enter module keywords: demo
Enter module source folder: src => demo

Existing Web Applications:
    /csp/user
    Enter a comma separated list of web applications or * for all:
Dependencies:
    Enter module:version or empty string to continue:
zpm:USER>
```

Note:

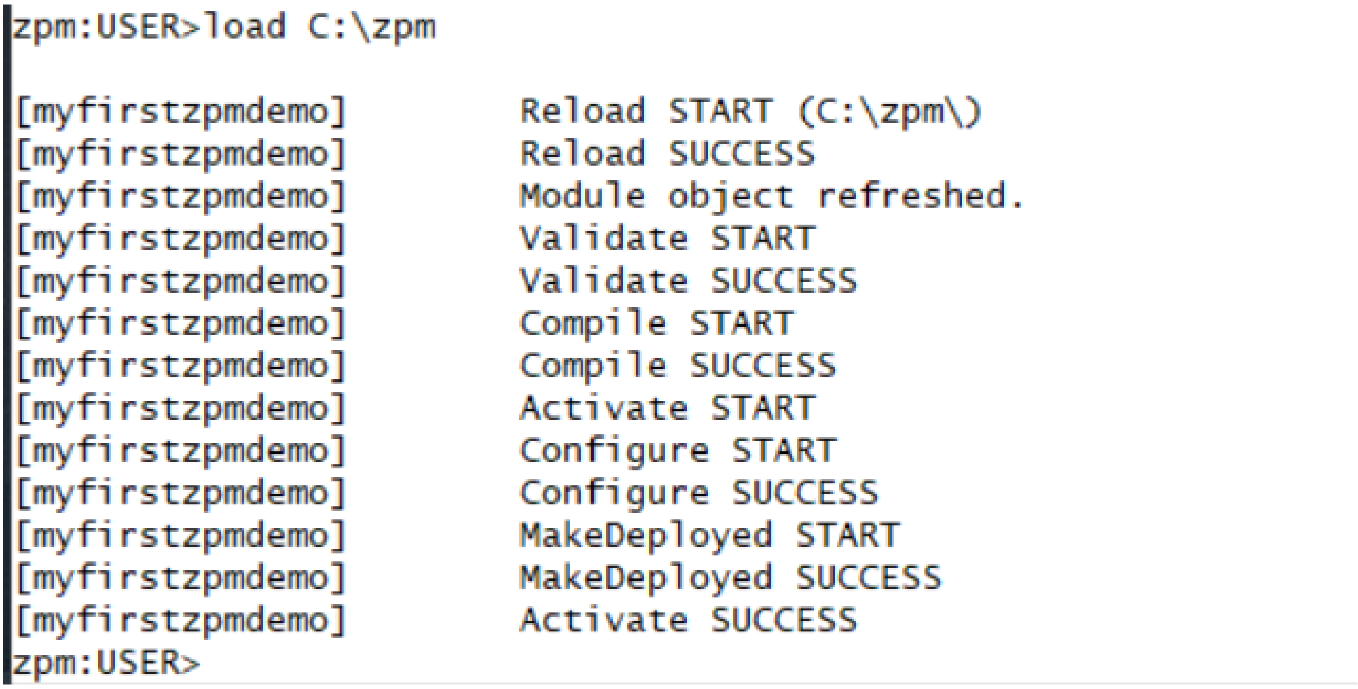
1. module version is using semantic versioning
2. module source folder is the folder has all the class file
3. zpm also offers a option to add web applications and dependency, in this example I will leave it blank

Now, open the file explorer, you will see a file called "module.xml" as you can see from the screenshot below



```xml
<?xml version="1.0" encoding="UTF-8"?>
<Export generator="Cache" version="25">
  <Document name="myfirstzpmdemo.ZPM">
    <Module>
      <Name>myfirstzpmdemo</Name>
      <Version>0.1.0</Version>
      <Description>this is my first zpm module</Description>
      <Keywords>demo</Keywords>
      <Packaging>module</Packaging>
      <Resource Name="Simple.PKG"/>
      <SourcesRoot>demo</SourcesRoot>
    </Module>
  </Document>
</Export>
```

Type command *load C:\ZPM\* you will see your module has been reloaded, validated, compiled and activated



```
zpm:USER>load C:\zpm

[myfirstzpmdemo]        Reload START (C:\zpm\)
[myfirstzpmdemo]        Reload SUCCESS
[myfirstzpmdemo]        Module object refreshed.
[myfirstzpmdemo]        Validate START
[myfirstzpmdemo]        Validate SUCCESS
[myfirstzpmdemo]        Compile START
[myfirstzpmdemo]        Compile SUCCESS
[myfirstzpmdemo]        Activate START
[myfirstzpmdemo]        Configure START
[myfirstzpmdemo]        Configure SUCCESS
[myfirstzpmdemo]        MakeDeployed START
[myfirstzpmdemo]        MakeDeployed SUCCESS
[myfirstzpmdemo]        Activate SUCCESS
zpm:USER>
```

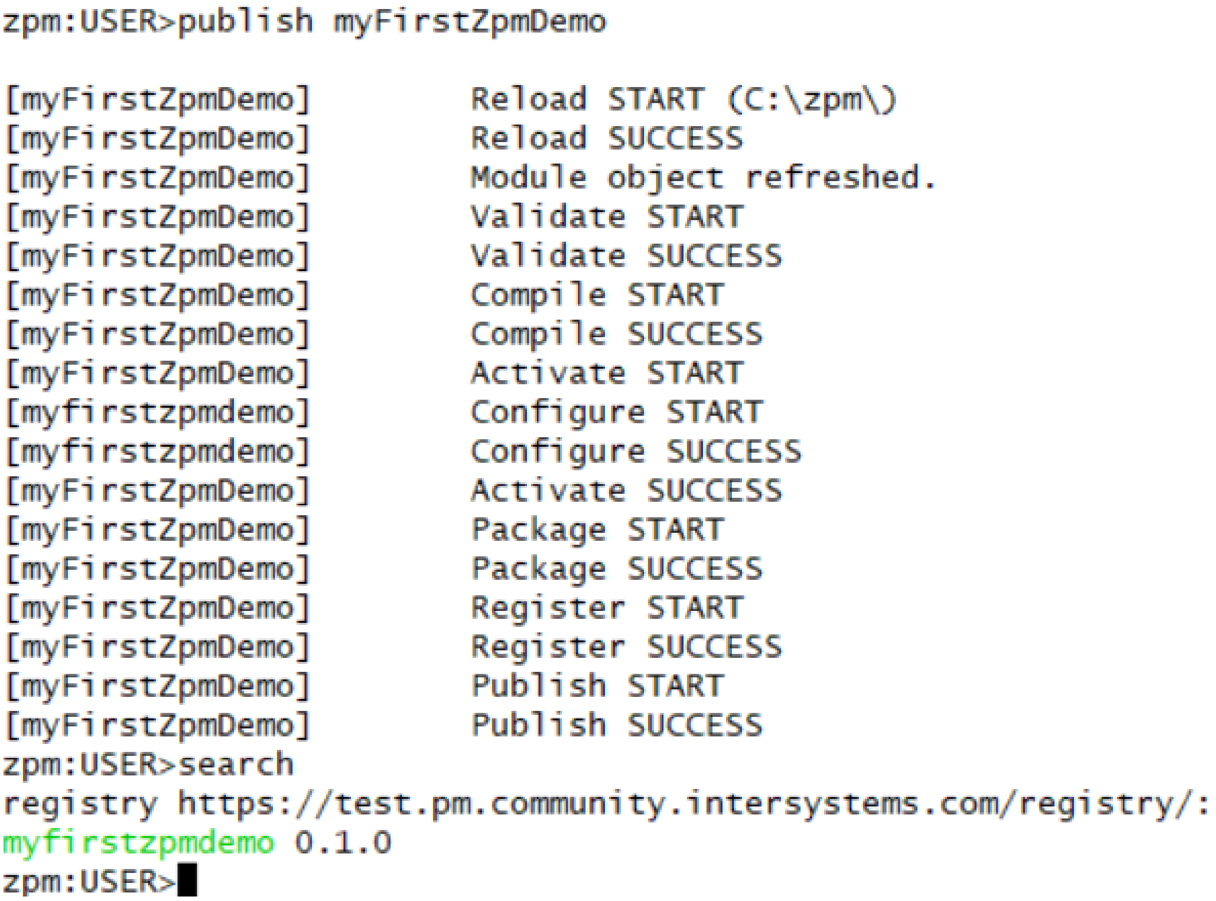
3. Find, Install, publish modules inside the Registry

Find packages available in the current Registry: *zpm:USER>search*

Install package from current Registry as an example lets install a module called zpmshow in public Registry: *zpm:USER>install zpmshow* (the command is install "moduleName")

Publish module after loaded: *zpm:USER>publish myFirstZPMDemo*

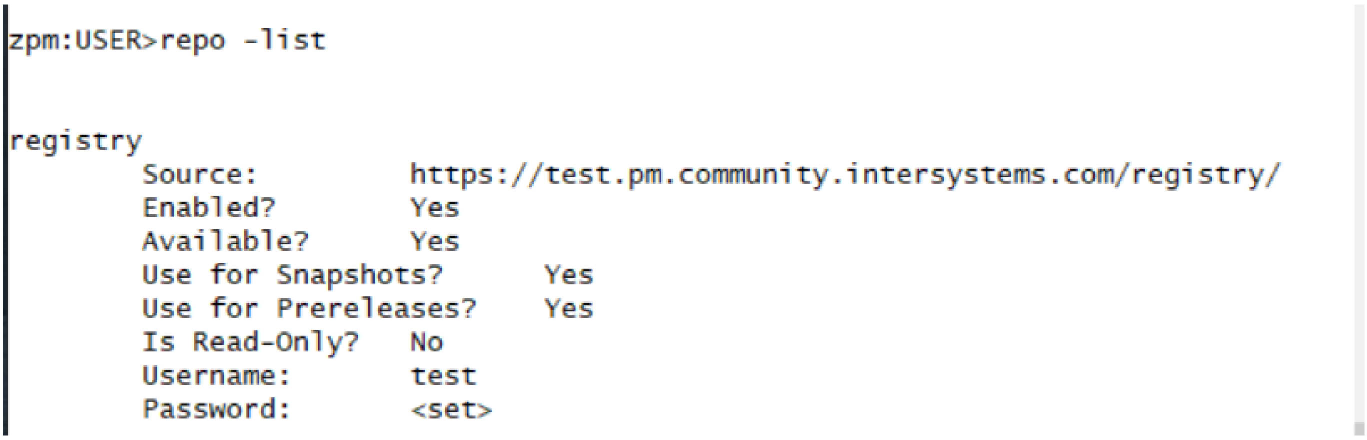
You can use *zpm:USER>search* to verify the publish, in my case you can see "myfirstzpmdemo 0.1.0" is sitting in current Registry.



```
zpm:USER>publish myFirstZpmDemo

[myFirstZpmDemo]        Reload START (C:\zpm\)
[myFirstZpmDemo]        Reload SUCCESS
[myFirstZpmDemo]        Module object refreshed.
[myFirstZpmDemo]        Validate START
[myFirstZpmDemo]        Validate SUCCESS
[myFirstZpmDemo]        Compile START
[myFirstZpmDemo]        Compile SUCCESS
[myFirstZpmDemo]        Activate START
[myfirstzpmdemo]        Configure START
[myfirstzpmdemo]        Configure SUCCESS
[myFirstZpmDemo]        Activate SUCCESS
[myFirstZpmDemo]        Package START
[myFirstZpmDemo]        Package SUCCESS
[myFirstZpmDemo]        Register START
[myFirstZpmDemo]        Register SUCCESS
[myFirstZpmDemo]        Publish START
[myFirstZpmDemo]        Publish SUCCESS
zpm:USER>search
registry https://test.pm.community.intersystems.com/registry/:
myfirstzpmdemo 0.1.0
zpm:USER>
```

Note: If you have an error when you are publishing a module that says: *"ERROR! Publishing module, something went wrong"*, make sure that the status of the current Registry is enabled and available.

You can use *zpm:USER>repo -list*, to verify the status of current Registry.



```
zpm:USER>repo -list


registry
        Source:         https://test.pm.community.intersystems.com/registry/
        Enabled?        Yes
        Available?      Yes
        Use for Snapshots?      Yes
        Use for Prereleases?    Yes
        Is Read-Only?   No
        Username:       test
        Password:       <set>
```

Video available: Click here

#Best Practices #InterSystems Package Manager (IPM) #InterSystems IRIS #InterSystems IRIS for Health

Source URL:https://community.intersystems.com/post/zpm-simple-implementation-cookbook