Highlight Note

Registering and Reusing Environments

After you've created an environment, you can register it in your workspace and reuse it for future experiments that have the same Python dependencies.

Registering an Environment

Use the **register** method of an **Environment** object to register an environment:

```
env.register(workspace=ws)
```

You can view the registered environments in your workspace like this:

```
from azureml.core import Environment

env_names = Environment.list(workspace=ws)
for env_name in env_names:
    print('Name:',env_name)
```

Retrieving and using an Environment

You can retrieve a registered environment by using the **get** method of the **Environment** class, and then assign it to a **ScriptRunConfig**.

For example, the following code sample retrieves the *training_environment* registered environment, and assigns it to an estimator:

```
from azureml.core import ScriptRunConfig, Environment

training_env = Environment.get(workspace=ws, name='training_environment')

script_config = ScriptRunConfig(source_directory='my_dir',
                                script='script.py',
                                environment=training_env)
```

When an experiment based on the estimator is run, Azure Machine Learning will look for an existing environment that matches the definition, and if none is found a new environment will be created based on the registered environment specification.

Curated Environments

Azure Machine Learning includes a selection of pre-defined *curated* environments that reflect common usage scenarios. These include environments that are pre-configured with package dependencies for common frameworks, such as Scikit-Learn, PyTorch, Tensorflow, and others.

Curated environments are registered in all Azure Machine Learning workspaces with a name that begins ***AzureML-***.

Note: You can't register your own environments with an “AzureML-” prefix.

To view curated environments and the dependencies they contain, you can run the following code:

```
from azureml.core import Environment

envs = Environment.list(workspace=ws)
for env in envs:
    if env.startswith("AzureML"):
        print("Name",env)
        print("packages", envs[env].python.conda_dependencies.serialize_to_string())
```