

FAQ

Listed below are a number of common issues users face with the various parts of the C++ API.

C++ Extensions

Undefined symbol errors from PyTorch/ATen

Problem: You import your extension and get an `ImportError` stating that some C++ symbol from PyTorch or ATen is undefined. For example:

```
>>> import extension
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: /home/user/.pyenv/versions/3.7.1/lib/python3.7/site-packages/extension.cpython-37m-x86_64-linux-gnu.so: undefined symbol: _Z
```

Fix: The fix is to `import torch` before you import your extension. This will make the symbols from the PyTorch dynamic (shared) library that your extension depends on available, allowing them to be resolved once you import your extension.

I created a tensor using a function from `at::` and get errors

Problem: You created a tensor using e.g. `at::ones` or `at::randn` or any other tensor factory from the `at::` namespace and are getting errors.

Fix: Replace `at::` with `torch::` for factory function calls. You should never use factory functions from the `at::` namespace, as they will create tensors. The corresponding `torch::` functions will create variables, and you should only ever deal with variables in your code.