Problem

Why is [the linker | the library loader at runtime] complaining about an undefined symbol which I declared as static member of one of my classes?

Solution

You forgot to instanciate the static member.

Whenever you declare a static member in one of your classes, then this declaration servers as a "place holder" telling the compiler/linker that this object will be defined elsewhere.

Normally you will declare the static member in the include file and define it, i.e. allocate memory for it, in a `cpp` file.

Example:

Header with declaration (foo.h):

```
class Foo
{
  private:
    static boolean active;
};
```

Implementation with definition (foo.cpp):

```
#include <foo.h>

boolean Foo::active(false);
```

There is a way to tell which symbols are undefined:

- Run
  ```
  nm --demangle=auto -u [YOUR_LIBRARY | YOUR_OBJECT_FILE]
  ```
  to detect all undefined symbols.

-- ThomasJuerges - 30 Nov 2006

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?