



Your Quill Toolbox: Delay Initialization

December 2025

DelayInit: Overview

- Defers “initialization” to a defined point in main.
- Implemented as a pointer daisy chain.
 - Side effect... multiple will execute upward in a source file
- Call `DelayInit::init_all()`
 - Meta does this with its “init”
 - Can be called repeatedly; all registered code called ONCE.
- Use macro `YQ_INVOKE` to register initialization code
 - Best practice is to create small local function, call that within the macro.

DelayInit: Registering Stuff

- Source Files A...Z

```
static void register_stuff()
{
    // register type, loader, etc
}

// Note the semicolon...
YQ_INVOKE(register_stuff());
```

- Called ONCE

- This code will only be called ONCE at program startup, in single threaded mode.

- Recommendations

- Private Function
 - Sub-functions if complicated
 - Note: Objects check for a `init_meta()` routine, will call that.

DelayInit: Main

- In Main

```
int main(int argc, char* argv[])
{
    // not using meta
    DelayInit::init_all( /* true */ );

    // using meta
    Meta::init();
    Meta::freeze();
}
```

- Initialization

- DelayInit::init_all()
 - Not using meta
- Meta::init()
 - Using meta (most apps)
- Repeat as libraries are loaded
(ie. plugins)