

---

Subject: Allegro5 - Scope addon

Posted by [Sender Ghost](#) on Wed, 22 Aug 2012 19:11:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

There are some ALLEGRO\_\* types, which require resource management, such as ALLEGRO\_BITMAP, ALLEGRO\_DISPLAY, ALLEGRO\_EVENT\_QUEUE with al\_destroy\_bitmap, al\_destroy\_display, al\_destroy\_event\_queue function respectively, etc.

For such cases I created allegro5/scope C++ addon. It consist of:

- al\_destroy template function, which invokes appropriate al\_destroy\_\* function, based on argument type:

Toggle example#include <allegro5/allegro.h>

#include <allegro5/allegro\_scope.h>

```
int main(int argc, const char *argv[]) {
    if (!al_init()) return 1;
    ALLEGRO_EVENT_QUEUE *queue = al_create_event_queue();
    ALLEGRO_DISPLAY *display = al_create_display(640, 480);
    ALLEGRO_BITMAP *bitmap = al_create_bitmap(32, 32);

    al_register_event_source(queue, al_get_display_event_source(display));

    al_destroy(queue);
    al_destroy(bitmap);
    al_destroy(display);

    return 0;
}
```

- Scope template class, which uses al\_destroy template function on scope completion:

Toggle example

#include <allegro5/allegro.h>

#include <allegro5/allegro\_scope.h>

```
int main(int argc, const char *argv[]) {
    if (!al_init()) return 1;
    Scope<ALLEGRO_EVENT_QUEUE> queue = al_create_event_queue();
    Scope<ALLEGRO_DISPLAY> display = al_create_display(640, 480);
    Scope<ALLEGRO_BITMAP> bitmap(al_create_bitmap(32, 32));

    al_register_event_source(~queue, al_get_display_event_source(~display));

    return 0;
}
```

- ScopeList class, which has container to store ALLEGRO\_\* types (actually, as void \*) and invoke appropriate al\_destroy\_\* function (in some predefined order, e.g. al\_destroy\_display function used last) on scope completion:

Toggle example

```
#include <allegro5/allegro.h>
```

```
#include <allegro5/allegro_scope.h>
```

```
int main(int argc, const char *argv[]) {  
    if (!al_init()) return 1;  
    ScopeList sl;  
    ALLEGRO_EVENT_QUEUE *queue = al_create_event_queue(); sl.Add(queue);  
    ALLEGRO_DISPLAY *display = sl << al_create_display(640, 480);  
    ALLEGRO_BITMAP *bitmap = sl << al_create_bitmap(32, 32);  
  
    al_register_event_source(queue, al_get_display_event_source(display));  
  
    return 0;  
}
```

Some notes:

The allegro5/allegro\_scope.h header file is context sensitive, which means, that you need to include it after other allegro\* header files (including addons). This is because of preprocessor, which checks used header files (by already defined header guards, e.g. `__al_included_allegro5_allegro_h`) to associate their `ALLEGRO_*` types and `al_destroy_*` functions (where associations created manually).

How to install:

The allegro5/scope is header-only addon, at the moment, with some defined directory structure for Allegro5 addons. I created allegro5/scope/scope.cpp file as a stub for compatibility with TheIDE, but not required to compile. Therefore, either:

- include the path to base folder with allegro5/scope addon

or

- in terms of TheIDE, create new assembly with following package nests:

AllegroDev;Allegro;Libraries;upp\uppsrc

where AllegroDev is path to base folder with allegro/scope addon; other paths explained above.

And add allegro5/scope package to your application package.

In the attachments archive you could find allegro5/scope addon and some example.

## File Attachments

1) [Allegro\\_v5\\_scope\\_addon.zip](#), downloaded 401 times

---