

```

/* dynamic loading: resolved at runtime */

fn fn_prototype(value i32) void

fn fn_load_symbol(lib_name string, symbol_name string) u32, error {
    err error

    handle := LoadLib(lib_name)

    fn_prototype, err = LoadSymbol(handle, symbol_name)

    if (err) ret error

    ret handle
}

/* dynamic loading with reflection */

@Module("module_name")
fn fn_prototype(value i32) void

fn fn_load_symbol(lib_name string, symbol_name string) u32, error {
    err error; get_module_name FnGetModuleName

    handle := LoadLib(lib_name)

    get_module_name, err = LoadSymbol(handle, "GetLlangModuleName")

    /**
     * "GetLlangModuleName" is predefined function to get the module name.
     * It is created by the compiler when you compile a dynamic library
     */

    module_name string = get_module_name()

    global_vars := Reflection.getGlobalVariables

    loop var : global_vars {
        if var.hasAnnotation("Module") {
            annot := cast(@Module, var.getFirstAnnotation("Module"))
            if annot.module_name == module_name {
                var.set_value, err = LoadSymbol(handle, var.name)
                if (err) {
                    ReportError("Module '${module_name}' has not symbol '${var.name}'")
                }
            }
        }
    }

    if err {
        ret error
    }

    ret handle
}

```