



TEXAS ADVANCED COMPUTING CENTER

WWW.TACC.UTEXAS.EDU



TEXAS

The University of Texas at Austin

Lmod hooks

Robert McLay

Sept. 7, 2021

Lmod Hooks Discussion



- ▶ History: How to track module usage?
- ▶ How do hooks work?
- ▶ What Hooks are available?

How to track module usage?

- ▶ No extra code in modulefiles.
- ▶ Must work with modulefiles written in Lua or TCL.
- ▶ Must not be built-in to Lmod.
- ▶ \Rightarrow Hooks

Hooks

- ▶ I am an emacs user
- ▶ Is there any other editor worth using?
- ▶ Emacs has hooks to modify behavior
- ▶ Hooks and SitePackage.lua was born.

SitePackage.lua

- ▶ SitePackage.lua is “sourced” on every module command.
- ▶ It provides functions available for every Lua module file.
- ▶ Site should set `$LMOD_PACKAGE_PATH` to the directory containing it.
- ▶ See `sandbox_registration{}` to see how to add your own functions.

Hooks

- ▶ Functions name can be stored in variable, arrays, etc.
- ▶ In particular, Lua can store strings and a matching function
- ▶ Lmod has default implementations that do nothing
- ▶ Site can override hooks with their own implementation.

Hooks in SitePackage.lua

```
local function load_hook(t)
    -- ...
    -- ...
end

local hook = require("hook")
hook.register("load", load_hook)
```

What can a site do with a load hook?

- ▶ The load hooks is called on every load.
- ▶ TACC uses the load hook to track module usage
- ▶ Many EasyBuild sites track only “user” loads
- ▶ Compute Canada track usage, set properties, and much more

Contrib/more_hooks/SitePackage.lua

```
local frameStk = FrameStk:singleton()
local userload = (frameStk:atTop()) and
                  "yes" or "no"

local logTbl      = {}
logTbl[#logTbl+1] = {"userload", userload}
logTbl[#logTbl+1] = {"module", t.modFullName}
logTbl[#logTbl+1] = {"fn", t.fn}

logmsg(logTbl)
```

Compute Canada

- ▶ <https://github.com/ComputeCanada/software-stack-config/blob/main/lmod/SitePackage.lua#L261-L272>

```
local function load_hook(t)
  local valid = validate_license(t)
  set_props(t)
  set_family(t)
  default_module_change_warning(t)
  log_module_load(t, true)
  set_local_paths(t)
end
```

- ▶ Setting families: https://github.com/ComputeCanada/software-stack-config/blob/main/lmod/SitePackage_families.lua
- ▶ Logging code: https://github.com/ComputeCanada/software-stack-config/blob/main/lmod/SitePackage_logging.lua
- ▶ Module properties
https://github.com/ComputeCanada/software-stack-config/blob/main/lmod/SitePackage_properties.lua
- ▶ https://github.com/ComputeCanada/software-stack-config/blob/main/lmod/SitePackage_visible.lua

What hooks are available?

- ▶ Basic hooks
- ▶ Shared home Filesystem hook
- ▶ Advanced Hooks

Basic hooks

- ▶ load hook - Called on every load
- ▶ avail - Map directories into labels
- ▶ startup - called at startup
- ▶ finalize - called just before Lmod is about to finish
- ▶ isVisibleHook - Reports whether a module should be visible to avail and spider.
- ▶ SiteName - Hook to specify Site Name
(*site_FAMILY_something*) for family prefix for hierarchical module layout: TACC_FAMILY_COMPILER

isVisibleHook

- ▶ isVisibleHook(modT): Reports whether a module should be visible to avail and spider.
- ▶ modT={fullName=..., sn=..., fn=...}
- ▶ Where fullName is name/version, sn = name and fn is fileName

Shared Home Filesystem hook

- ▶ `groupName` - This hook adds the arch and os name to `moduleT.lua` to make it safe on shared filesystems.

Advanced Hooks

- ▶ unload - Called on every unload
- ▶ msgHook - Hook to print message after avail, list, spider
- ▶ errWarnMsgHook - Hook to print messages after LmodError, LmodWarning, LmodMessage
- ▶ restore - Hook to run after a restore operation.
- ▶ load_spider - This hook is run for evaluating modules for spider/avail.
- ▶ listHook - This hook gets the list of active modules
- ▶ spider_decoration - This hook adds decoration to spider level one output. It can be the category or a property.

HookArray

- ▶ A module can load other modules
- ▶ Module X loads modules A, B, and C
- ▶ Suppose that A, B load and C errors out.
- ▶ Use `load_hook()` to save loads in an array
- ▶ Use the HookArray to report loads via syslog.

Contrib/TACC/SitePackage.lua:

```
local s_msgA = {}
local function load_hook(t)
    -- the arg t is a table:
    --      t.modFullName:  the module full name: (i.e: gcc/4.7.2)
    --      t.fn:           The file name: (i.e /apps/mfiles/Core/gcc/4.7.2.lua)
    if (mode() ~= "load") then return end
    local currentTime = epoch()
    local msg          = string.format("user=%s module=%s path=%s host=%s time=%f",
                                      user, t.modFullName, t.fn,
                                      host or "<unknown_syshost>", currentTime)

    local a            = s_msgA
    a[#a+1]            = msg
    dbg.fini()
end
local function report_loads()
    local a = s_msgA
    for i = 1,#a do
        local msg = a[i]
        lmod_system_execute("logger -t ModuleUsageTracking -p local0.info " .. msg)
    end
end

ExitHookA.register(report_loads)
hook.register("load",          load_hook)
```

Other hooks in the documentation

- ▶ Avail:
https://lmod.readthedocs.io/en/latest/200_avail_custom.html
- ▶ msgHook:
https://lmod.readthedocs.io/en/latest/170_hooks.html