

```

function luatexbase.reset_callback(name, make_false) for, vinpairs (luatexbase.callback_descriptions
truethen luatexbase.disable_callback(name) endend luatexbase.base_add_callback =
luatexbase.add_callback function luatexbase.add_callback(name, fun, description, priority) local
priority if priority == nil then priority = #luatexbase.callback_descriptions(name) +
1 end if (luatexbase.callback_types[name] == 3 and priority == 1 and #luatexbase.callback_descript
1) then luatexbase.module_warning("luatexbase", "resetting exclusive callback :
"..name) luatexbase.reset_callback(name) end local saved_callback = , ff, dd fork, vinpairs (luatexba
priority then ff, dd = luatexbase.remove_from_callback(name, v) saved_callback[k] =
ff, dd endend luatexbase.base_add_callback(name, fun, description) fork, vinpairs (saved_callback
setmetatable(['latex - package'] = 0, ini = 0, string = 0, other = 0, latex = 0, expl = 0, expl3 = 0)

local whatsitdata = user_whatsits[package] if not whatsitdata then whatsitdata =
user_whatsits[package] = whatsitdata end

local id = whatsitdata[name] if id then warning("replacing whatsit %s:%s
(%d)", package, name, id) else id = luatexbase.new_whatsit(name) whatsitdata[name] =
id whatsit_ds[id] = name, package end return id end luatexbase.new_user_whatsit_id =
new_user_whatsit_id local new_user_whatsit = function(req, package) local id, whatsit_if_type(req) ==
"string" then id = new_user_whatsit_id(req, package) whatsit = newnode(whatsit_t, user_defined_t) wh
id else if req.id == whatsit_t and req.subtype == user_defined_t then id = req.user_id whatsit =
copynode(req) if not whatsit_ds[id] then warning("whatsit id %d unregistered; .." inconsistencies m
new_user_whatsit local get_user_whatsit_id = function(name, package) if not package then package =
unassociated end return user_whatsits[package][name] end luatexbase.get_user_whatsit_id =
get_user_whatsit_id local get_user_whatsit_name = function(asked) local id if type(asked) ==
"number" then id = asked else if type(asked) == "function" then local n =
asked() id = n.user_id else id = asked.user_id end local metadata = whatsit_ds[id] if not metadata then
get_user_whatsit_name local dump_registered_whatsits = function(asked_package) local whatsit_ist =
if asked_package then local whatsitdata = user_whatsits[asked_package] if not whatsitdata then error(
1) = stringformat("(%s : %s %d)", asked_package, name, id) end else texpwritenl("userwhatsita
1) = stringformat("(%s : %s %d)", package, name, id) end endend texpwritenl("local first =
true for i = 1, #whatsit_ist do if first then first = false else texpwritenl"" end texpwrite(whatsit
dump_registered_whatsits luatexbase.new_attribute = new_attribute luatexbase.new_user_whatsit =
new_user_whatsit luatexbase.new_user_whatsit_id = new_user_whatsit_id luatexbase.get_user_whatsit_id =
get_user_whatsit_id luatexbase.get_user_whatsit_name = get_user_whatsit_name luatexbase.dump_registered
dump_registered_whatsits

Hello from standard LATEX!
tex.print("Hello, World! from Lua")
hello

```