

Toggle menu  
Blue Gold Program Wiki

## Navigation

- [Main page](#)
- [Recent changes](#)
- [Random page](#)
- [Help about MediaWiki](#)

## Tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Permanent link](#)
- [Page information](#)

## Personal tools

- [Log in](#)

## personal-extra

Toggle search

Search

Random page

## Views

- [View](#)
- [View source](#)
- [History](#)
- [PDF Export](#)

## Actions

# Module:Unsubst

From Blue Gold Program Wiki

The printable version is no longer supported and may have rendering errors. Please update your browser bookmarks and please use the default browser print function instead.

*Documentation for this module may be created at [Module:Unsubst/doc](#)*

```

local checkType = require('libraryUtil').checkType

local p = {}

local BODY_PARAM = '$B'

local specialParams = {
    ['$params'] = 'parameter list',
    ['$aliases'] = 'parameter aliases',
    ['$flags'] = 'flags',
    ['$B'] = 'template content'
}

function p.main(frame, body)
    -- If we are substing, this function returns a template invocation,
    and if
    -- not, it returns the template body. The template body can be
    specified in
    -- the body parameter, or in the template parameter defined in the
    -- BODY_PARAM variable. This function can be called from Lua or from
    -- #invoke.

    -- Return the template body if we aren't substing.
    if not mw.isSubsting() then
        if body ~= nil then
            return body
        elseif frame.args[BODY_PARAM] ~= nil then
            return frame.args[BODY_PARAM]
        else
            error(string.format(
                "no template content specified (use parameter
                '%s' from #invoke)",
                BODY_PARAM
            ), 2)
        end
    end

    -- Sanity check for the frame object.
    if type(frame) ~= 'table'
        or type(frame.getParent) ~= 'function'
        or not frame:getParent()
    then
        error(
            "argument #1 to 'main' must be a frame object with a
            parent " ..
            "frame available",
            2
        )
    end

    -- Find the invocation name.

```

```

local mTemplateInvocation = require('Module:Template invocation')
local name = mTemplateInvocation.name(frame:getParent():getTitle())

-- Combine passed args with passed defaults
local args = {}
if string.find( '..',(frame.args['$flags'] or '')..' ','',
',%s*override%s*', ' ) then
    for k, v in pairs( frame:getParent().args ) do
        args[k] = v
    end
    for k, v in pairs( frame.args ) do
        if not specialParams[k] then
            if v == '__DATE__' then
                v =
mw.getContentLanguage():formatDate( 'F Y' )
            end
            args[k] = v
        end
    end
end
else
    for k, v in pairs( frame.args ) do
        if not specialParams[k] then
            if v == '__DATE__' then
                v =
mw.getContentLanguage():formatDate( 'F Y' )
            end
            args[k] = v
        end
    end
    for k, v in pairs( frame:getParent().args ) do
        args[k] = v
    end
end

-- Trim parameters, if not specified otherwise
if not string.find( '..',(frame.args['$flags'] or '')..' ','',
',%s*keep%-whitespace%s*', ' ) then
    for k, v in pairs( args ) do args[k] = mw.ustring.match(v,
'^%s*(.*)%s*$') or '' end
end

-- Pull information from parameter aliases
local aliases = {}
if frame.args['$aliases'] then
    local list = mw.text.split( frame.args['$aliases'], '%s*,%s*'
)
    for k, v in ipairs( list ) do
        local tmp = mw.text.split( v, '%s*>%s*' )
        aliases[tonumber(mw.ustring.match(tmp[1],
'^[1-9][0-9]*$')) or tmp[1]] = ((tonumber(mw.ustring.match(tmp[2],
'^[1-9][0-9]*$')) or tmp[2]))
    end
end

```

```

        end
    end
    for k, v in pairs( aliases ) do
        if args[k] and ( not args[v] or args[v] == '' ) then
            args[v] = args[k]
        end
        args[k] = nil
    end

    -- Remove empty parameters, if specified
    if string.find( ',,..(frame.args['$flags'] or '..'),', ', '%s*remove%-
empty%s*', ' ) then
        local tmp = 0
        for k, v in ipairs( args ) do
            if v ~= '' or ( args[k+1] and args[k+1] ~= '' ) or (
args[k+2] and args[k+2] ~= '' ) then
                tmp = k
            else
                break
            end
        end
        for k, v in pairs( args ) do
            if v == '' then
                if not (type(k) == 'number' and k < tmp) then
args[k] = nil end
            end
        end
    end

    -- Order parameters
    if frame.args['$params'] then
        local params, tmp = mw.text.split( frame.args['$params'],
'%s*,%s*' ), {}
        for k, v in ipairs(params) do
            v = tonumber(mw.ustring.match(v, '^[1-9][0-9]*$')) or
v
            if args[v] then tmp[v], args[v] = args[v], nil end
        end
        for k, v in pairs(args) do tmp[k], args[k] = args[k], nil end
        args = tmp
    end

    return mTemplateInvocation.invocation(name, args)
end

p[''] = p.main -- For backwards compatibility

return p

```

Retrieved from "<https://www.bluegoldwiki.com/index.php?title=Module:Unsubst&oldid=232>"

## Namespaces

- [Module](#)
- [Discussion](#)

## Variants

This page was last edited on 18 February 2020, at 14:18.

## Blue Gold Program Wiki

The wiki version of the Lessons Learnt Report of the Blue Gold program, documents the experiences of a technical assistance (TA) team working in a development project implemented by the Bangladesh Water Development Board (BWDB) and the Department of Agricultural Extension (DAE) over an eight+ year period from March 2013 to December 2021. The wiki lessons learnt report (LLR) is intended to complement the BWDB and DAE project completion reports (PCRs), with the aim of recording lessons learnt for use in the design and implementation of future interventions in the coastal zone.

- [Privacy policy](#)
- [About Blue Gold Program Wiki](#)
- [Disclaimers](#)

Developed and maintained by Big Blue Communications for Blue Gold Program



[Blue Gold Program Wiki](#)