

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:File link

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:File link/doc](#)

```

-- This module provides a library for formatting file wikilinks.

local yesno = require('Module:Yesno')
local checkType = require('libraryUtil').checkType

local p = {}

function p._main(args)
    checkType('_main', 1, args, 'table')

    -- This is basically libraryUtil.checkTypeForNamedArg, but we are
rolling our
    -- own function to get the right error level.
    local function checkArg(key, val, level)
        if type(val) ~= 'string' then
            error(string.format(
                "type error in '%s' parameter of '_main'
(expected string, got %s)",
                    key, type(val)
                ), level)
        end
    end

    local ret = {}

    -- Adds a positional parameter to the buffer.
    local function addPositional(key)
        local val = args[key]
        if not val then
            return nil
        end
        checkArg(key, val, 4)
        ret[#ret + 1] = val
    end

    -- Adds a named parameter to the buffer. We assume that the parameter
name
    -- is the same as the argument key.
    local function addNamed(key)
        local val = args[key]
        if not val then
            return nil
        end
        checkArg(key, val, 4)
        ret[#ret + 1] = key .. '=' .. val
    end

    -- Filename
    checkArg('file', args.file, 3)
    ret[#ret + 1] = 'File:' .. args.file

```

```

-- Format
if args.format then
    checkArg('format', args.format)
    if args.formatfile then
        checkArg('formatfile', args.formatfile)
        ret[#ret + 1] = args.format .. '=' .. args.formatfile
    else
        ret[#ret + 1] = args.format
    end
end

-- Border
if yesno(args.border) then
    ret[#ret + 1] = 'border'
end

addPositional('location')
addPositional('alignment')
addPositional('size')
addNamed('upright')
addNamed('link')
addNamed('alt')
addNamed('page')
addNamed('class')
addNamed('lang')
addNamed('start')
addNamed('end')
addNamed('thumbtime')
addPositional('caption')

return string.format('[[[%s]]]', table.concat(ret, '|'))
end

function p.main(frame)
    local origArgs = require('Module:Arguments').getArgs(frame, {
        wrappers = 'Template:File link'
    })
    if not origArgs.file then
        error("'file' parameter missing from [[Template:File link]]",
0)
    end

    -- Copy the arguments that were passed to a new table to avoid
looking up
    -- every possible parameter in the frame object.
    local args = {}
    for k, v in pairs(origArgs) do
        -- Make _BLANK a special argument to add a blank parameter.
For use in
        -- conditional templates etc. it is useful for blank
arguments to be

```

```

-- ignored, but we still need a way to specify them so that
we can do
-- things like [[File:Example.png|link=]].
if v == '_BLANK' then
    v = ''
end
args[k] = v
end
return p._main(args)
end

return p

```

Retrieved from "https://www.bluegoldwiki.com/index.php?title=Module:File_link&oldid=2275"

Namespaces

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

Variants

This page was last edited on 26 April 2020, at 05:11.

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](#)