

☐ 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:Namespace detect

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:Namespace detect/doc](#)

[illegible]

```

        return nil
    end

    local function equalsArrayValue(t, value)
        -- Returns true if value equals a value in the array t. Otherwise
        -- returns false.
        for i, arrayValue in ipairs(t) do
            if value == arrayValue then
                return true
            end
        end
        return false
    end

end

function p.getPageObject(page)
    -- Get the page object, passing the function through pcall in case of
    -- errors, e.g. being over the expensive function count limit.
    if page then
        local success, pageObject = pcall(mw.title.new, page)
        if success then
            return pageObject
        else
            return nil
        end
    else
        return mw.title.getCurrentTitle()
    end
end

-- Provided for backward compatibility with other modules
function p.getParamMappings()
    return mappings
end

local function getNamespace(args)
    -- This function gets the namespace name from the page object.
    local page = fetchValue(args, argKeys.demopage)
    if page == '' then
        page = nil
    end
    local demospace = fetchValue(args, argKeys.demospace)
    if demospace == '' then
        demospace = nil
    end
    local subjectns = fetchValue(args, argKeys.subjectns)
    local ret
    if demospace then
        -- Handle "demospace = main" properly.
        if equalsArrayValue(argKeys.main, ustringLower(demospace))
then
            ret = mw.site.namespaces[0].name

```

```

        else
            ret = demospace
        end
    else
        local pageObject = p.getPageObject(page)
        if pageObject then
            if pageObject.isTalkPage then
                -- Get the subject namespace if the option is
set,
                -- otherwise use "talk".
                if yesno(subjectns) then
                    ret =
mw.site.namespaces[pageObject.namespace].subject.name
                else
                    ret = 'talk'
                end
            else
                ret = pageObject.nsText
            end
        else
            return nil -- return nil if the page object doesn't
exist.
        end
    end
    ret = ret:gsub('_', ' ')
    return ustringLower(ret)
end

function p._main(args)
    -- Check the parameters stored in the mappings table for any matches.
    local namespace = getNamespace(args) or 'other' -- "other" avoids nil
table keys
    local params = mappings[namespace] or {}
    local ret = fetchValue(args, params)
    --[[
    -- If there were no matches, return parameters for other namespaces.
    -- This happens if there was no text specified for the namespace that
    -- was detected or if the demospace parameter is not a valid
be
    -- namespace. Note that the parameter for the detected namespace must
    -- completely absent for this to happen, not merely blank.
    --]]
    if ret == nil then
        ret = fetchValue(args, argKeys.other)
    end
    return ret
end

function p.main(frame)
    mArguments = require('Module:Arguments')
    local args = mArguments.getArgs(frame, {removeBlanks = false})

```

```

    local ret = p._main(args)
    return ret or ''
end

function p.table(frame)
    --[[
    -- Create a wikitable of all subject namespace parameters, for
    -- documentation purposes. The talk parameter is optional, in case it
    -- needs to be excluded in the documentation.
    --]]
    -- Load modules and initialise variables.
    mTableTools = require('Module:TableTools')
    local namespaces = mw.site.namespaces
    local cfg = data.cfg
    local useTalk = type(frame) == 'table'
        and type(frame.args) == 'table'
        and yesno(frame.args.talk) -- Whether to use the talk
parameter.
    -- Get the header names.
    local function checkValue(value, default)
        if type(value) == 'string' then
            return value
        else
            return default
        end
    end
    local nsHeader = checkValue(cfg.wikitableNamespaceHeader,
'Namespace')
    local aliasesHeader = checkValue(cfg.wikitableAliasesHeader,
'Aliases')

    -- Put the namespaces in order.
    local mappingsOrdered = {}
    for nsname, params in pairs(mappings) do
        if useTalk or nsname ~= 'talk' then
            local nsid = namespaces[nsname].id
            -- Add 1, as the array must start with 1; nsid 0
would be lost otherwise.
            nsid = nsid + 1
            mappingsOrdered[nsid] = params
        end
    end
    mappingsOrdered = mTableTools.compressSparseArray(mappingsOrdered)

    -- Build the table.
    local ret = '{| class="wikitable"'
        .. '\n|-'
        .. '\n! ' .. nsHeader
        .. '\n! ' .. aliasesHeader
    for i, params in ipairs(mappingsOrdered) do
        for j, param in ipairs(params) do

```

```

        if j == 1 then
            ret = ret .. '\n|-'
                .. '\n| <code>' .. param .. '</code>'
                .. '\n| '
        elseif j == 2 then
            ret = ret .. '<code>' .. param .. '</code>'
        else
            ret = ret .. ', <code>' .. param .. '</code>'
        end
    end
end
ret = ret .. '\n|-'
    .. '\n|}'
return ret
end

return p

```

Retrieved from

"https://www.bluegoldwiki.com/index.php?title=Module:Namespace_detect&oldid=248"

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](#)