



# Template:Seealso

```
/***
```

```
  USAGE:
```

```
  SeeAlso(pages, exclude, limit, parents)
```

```
    Show pages related to current page by hierarchy and by see: and h
```

```
  PARAMETERS:
```

```
  (optional) pages : list (default: child pages of current page)
```

```
    List of pages to organize based on their tags.
```

```
  (optional) exclude : list (default: none)
```

```
    List of pages to exclude from the related list.
```

```
  (optional) limit : num (default: 8)
```

```
    Limit of related pages to use to determine if additional pages ne
```

```
  (optional) parents : bool (default: true)
```

```
    Show link to parent page if no other links are shown.
```

```
***/  
  
// Redirect to custom page, if any  
var template_custom = wiki.inclusions(true)[-1].path .. 'Custom';  
if (wiki.pageexists(template_custom)) {  
  return template(template_custom, $);  
}  
  
var pages = $0 ?? $pages;  
var exclude = $1 ?? $exclude ?? [ ];  
var limit = $2 ?? $limit ?? 5;  
var parents = $2 ?? $parents ?? true;  
  
//--- Aggregate Pages ---  
  
// if no pages were provided, enumerate child pages  
if(pages is nil) {  
  let pages = [  
    wiki.getpage(id)  
  ]  
  foreach  
    var id in [ num.cast(xml.text(id)) foreach var id in wiki.tre
```

```

    ];
}

// find tags with the 'see:' prefix (or no prefix) and include the definition
let pages ..= [
    p .. { origin: 'tags' }
    foreach
        var tag in page.tags where (tag.type === 'text') && (string.starts
        var def = tag.definition,
        var p in ((def && (def.id != page.id)) ? [ def ] : tag.pages) where
];

// find tags with the 'keyword:' prefix and include search results for the tag
if(#pages < limit) {
    var terms = [
        "" .. string.escape(string.substr(tag.name, 5)) .. ""
    ]
    foreach
        var tag in page.tags where (tag.type == 'text') && string.starts
];
if(#terms) {
    var searchString = string.join(terms, ' ');
    var defaultSearch = 'type:wiki AND namespace:main';

    // find main guide (tagged article:topic-guide), if any
    var pathSearch = '';
    foreach (var parent in page.parents) {
        var isTopic = #[tag foreach var tag in parent.tags where (tag.type == 'text') && string.starts
        if (isTopic) {
            let pathSearch = ' AND path:' .. parent.path .. '/*';
            break;
        }
    }

    // keyword must appear in title
    var titleSearch = ' AND title:' .. searchString;
    var searchQuery = defaultSearch .. titleSearch .. pathSearch;

    let pages ..= [
        p .. { origin: 'keyword' }
        foreach
            var p in wiki.getsearch(searchString, limit - #pages, _, _);
    ];
}
}

```

```

// de-duplicate pages
var uniquepages = [ ];
var pageids = [ ];
foreach(var p in pages) {
    if((p.id != page.id) && (p.id not in pageids) && (p.id not in exclude
        let pageids .= [ p.id ];
        let uniquepages .= [ p ];
    }
}
let pages = list.sort(uniquepages, 'path');

// enumerate all pages and classify them
let pages = [
    p .. {
        topic: #[ t foreach var t in tags where string.startswith(t.value
        tutorial: #[ t foreach var t in tags where string.startswith(t.va
        troubleshooting: #[ t foreach var t in tags where string.startswi
        reference: #[ t foreach var t in tags where string.startswith(t.v
        beginner: #[ t foreach var t in tags where string.endswith(t.valu
        intermediate: #[ t foreach var t in tags where string.endswith(t.
        advanced: #[ t foreach var t in tags where string.endswith(t.valu
    } foreach
        var p in pages where p.id != page.id,
        var tags = p.tags
];

```

```

<div class="noindex">
    <div class="mt-control-block-content">

        //--- Show Topic Pages ---
        var topics = [ p foreach var p in pages where p.topic ];
        if(#topics > 0) {
            if(#topics < #pages) {
                <div class="mt-control-page-subtitle"> wiki.localize("Mir
            }
            <dl class="mt-control-seealso">
                foreach(var p in topics) {
                    <dt> web.link(p.uri, p.title) </dt>
                    <dd> wiki.text(p.path, "Overview") </dd>
                }
            </dl>
        }

        //--- Show Tutorial Pages ---
        var tutorials = [ p foreach var p in pages where p.tutorial ];

```

```

var beginner_label = xml.text(wiki.localize("MindTouch.IDF.tutori
var intermediate_label = xml.text(wiki.localize("MindTouch.IDF.tu
var advanced_label = xml.text(wiki.localize("MindTouch.IDF.tutori
var tutorials = [ p .. { sortlabel: beginner_label } foreach var
    [ p .. { sortlabel: intermediate_label } foreach var p in tut
    [ p .. { sortlabel: advanced_label } foreach var p in tutoria
if(#tutorials > 0) {
    if(#tutorials < #pages) {
        <div class="mt-control-page-subtitle"> wiki.localize("Mir
    }
    template("ListPages", {
        pages: tutorials,
        style: "bullets",
        sort: "custom"
    });
}

//--- Show Troubleshooting Pages ---
var troubleshootings = [ p foreach var p in pages where p.trouble
if(#troubleshootings > 0) {
    if(#troubleshootings < #pages) {
        <div class="mt-control-page-subtitle"> wiki.localize("Mir
    }
    template("ListPages", {
        pages: troubleshootings,
        style: "bullets",
        sort: "custom"
    });
}

//--- Show Reference Pages ---
var references = [ p foreach var p in pages where p.reference ];
if(#references > 0) {
    if(#references < #pages) {
        <div class="mt-control-page-subtitle"> wiki.localize("Mir
    }
    template("ListPages", {
        pages: references,
        style: "bullets",
        sort: "custom"
    });
}

//--- Show Miscellaneous Pages ---
var misc = [ p foreach var p in pages where !p.topic && !p.tutori

```

```
if(#misc > 0) {
    if(#misc < #pages) {
        <div class="mt-control-page-subtitle"> wiki.localize("Miscellaneous")
    }
    template("ListPages", {
        pages: misc,
        style: "bullets",
        sort: "custom"
    });
}

//--- Show Parent Page (if no other pages were shown) ---
if(parents && !#topics && !#tutorials && !#troubleshootings && !#faq) {
    template("ListPages", {
        pages: [ page.parent ],
        style: "bullets",
        sort: "custom"
    });
}
</div>
</div>
```