authors, admins (advanced)

Using the (:if:) Directive

The `(:if:)` directive allows portions of a page to be included or excluded from processing. The generic form of the `(:if:)` directive is

```
(:if cond param:) body (:ifend:)
```

where "cond" names a condition to be tested (described below), and "param" is a parameter or other argument to the condition.

The built-in conditions include:

`(:if name PAGENAME:)`	-	current page is named "`PAGENAME`"
`(:if group GROUPNAME:)`	-	current group is named "`GROUPNAME`"
`(:if auth LEVEL PAGENAME:)`	-	viewer is authorized at "`LEVEL`" where `LEVEL` can be: `read`, `edit`, `upload`, `attr` or `admin`; `PAGENAME` is optional.
`(:if authid:)`	-	current viewer is authenticated
`(:if true:)`	-	always include text
`(:if false:)`	-	always exclude text (same as a comment)
`(:if attachments:)`	-	current page has attachments
`(:if date DATE:)`	-	true if current date is `DATE`
`(:if date DATE..:)`	-	true if current date is `DATE` or later (unlimited)
`(:if date DATE1..DATE2:)`	-	true if current date is in range `DATE1` to `DATE2` (inclusive)
	dates are in the format yyyy-mm-dd or yyyymmdd	
`(:if enabled VAR:)`	-	true if PHP VAR is true
`(:if enabled AuthPw:)`	-	true if user has entered any password during the current browser session.
`(:if equal STRING1 STRING2:)`	-	true if `STRING1` equals `STRING2`
`(:if match REG_EXPRESSION:)`	-	true if current page name matches the regular expression
`(:if exists PAGENAME:)`	-	true if the page *pagename* exists

Negated forms of conditions also work:

`(:if !attachments:)`	-	this page has no attachments

`(:if ! name PAGENAME:)`
`(:if name -PAGENAME :)`
`(:if name !PAGENAME :)`
		current page is NOT named "`PAGENAME`"

Any `(:if:)` automatically terminates the previous one, thus markup can be easily cased (and are not nested):

```
(:if enabled AuthPw:)* You're logged in
 (:if auth read:)* You can read
 (:if auth edit:)* You can edit
 (:if auth upload:)* You can upload
 (:ifend:)
```

Using wildcard placeholders

The character `*` can be used as a wildcard to represent any character, zero, one or multiple times.
The character ? can be used as a wildcard to represent any character exactly one time.
Wildcard characters (`*` and ?) can be used with the *name* and *group* conditional markups, thus:

`(:if name PmCal.2005* :)`	-	current page is in group PmCal and begins with 2005
`(:if group PmWiki* :)`	-	current page is in group PmWiki or a group beginning with Pmwiki
`(:if name Profiles.*,-Profiles.Profiles :)`	-	current page is in group `Profiles` but not `Profiles.Profiles`

Combining conditions

Conditions (as previously defined) may be combined into more complex conditional expressions using one of these three equivalent forms:

```
(:if expr EXPRESSION :)
 (:if [ EXPRESSION ] :)
 (:if ( EXPRESSION ) :)
```

Conditions are combined into expressions with boolean operators and brackets. In the next table, A and B are either regular conditions or (round-)bracketed sub-expressions of regular conditions:

Expression	Operator	Result
`A and B`	And	TRUE if both A and B are TRUE.
`A or B`	Or	TRUE if either A or B is TRUE.
`A xor B`	Xor	TRUE if either A or B is TRUE, but not both.
`! A`	Not	TRUE if A is not TRUE.
`A && B`	And	TRUE if both A and B are TRUE.

A || B	Or	TRUE if either A or B is TRUE.

Nota:

- Spaces around operators and brackets are required.
- No specific feedback is given for syntaxic errors or unbalanced brackets.
- Use round brackets (not square) for nested expressions.

Thus, the following is a valid way of building an expression that shows the following contents only when the user is either the administrator, or is logged in and the time is later than the given date:

```
(:if [ auth admin || ( authid && date 2006-06-01 ) ] :)
```

Nesting with square brackets will silently fail to work as expected:

`(:if [ auth admin || [ authid && date 2006-06-01 ] ] :)` NOTE: Doesn't Work!

A common use of these complex tests are for expressions like:

```
(:if expr auth admin || auth attr || auth edit :)
[[Logout -> {$Name}?action=logout]]
(:if:)
```

which provides you a *logout* link *only* when authentified with rights higher than 'read'.

admins (advanced)

Creating new conditions

See Cookbook:ConditionalMarkupSamples.

<< InterMap | DocumentationIndex | Page variables >>