exp:template:check: rule reference

Source file: doc/eng-us/reference/template-check-rules.md Last changed 2026-11-04
https://doc.exponential.earth/learninminutes/exphelp/viewer/reference/template-check-rules.md

Every finding of exp:template:check carries a rule id. This page lists every rule: its severity, the level or option that turns it on, what it means, and an example of a template that triggers it with the fix. `./console exp:template:check --rules` prints the list, `--explain=<id>` one rule.

Rule ids are permanent: a rule may be retired (then it never fires again), but its id is never reused (Compatibility promise).

The outputs below come from running the check on a development installation. The machine output is described in the JSON schema; see The machine output at the end of this page.

Rule	Severity	Engine	Turned on by
TPL-SYNTAX-001	ERROR	tpl	every level
TPL-SYNTAX-002	WARN	tpl	every level
TPL-COMPILE-001	ERROR	tpl	`compile`, `strict`
TPL-STRICT-INCLUDE-MISSING	ERROR	tpl	`strict`, `--includes`
TPL-STRICT-UNKNOWN-OPERATOR	WARN	tpl	`strict`
STPL-SYNTAX-001	ERROR	stpl	every level
STPL-COMPILE-001	ERROR	stpl	`compile`, `strict`
STPL-SECURITY-001	ERROR	stpl	every level, with `smarty.ini Security=enabled`
STPL-STRICT-INCLUDE-MISSING	ERROR	stpl	`strict`, `--includes`
TWIG-SYNTAX-001	ERROR	twig	every level
TWIG-COMPILE-001	ERROR	twig	`compile`, `strict`
TWIG-SECURITY-001	ERROR	twig	`strict`, `--security`
TWIG-STRICT-UNKNOWN-OPERATOR	WARN	twig	`strict`
TWIG-STRICT-INCLUDE-MISSING	ERROR	twig	`strict`, `--includes`
TWIG-DEPRECATION-001	WARN	twig	`--deprecations`
ENCODING-UTF8-001	ERROR	all	`--encoding`
ENCODING-BOM-002	WARN	all	`--encoding`
ENCODING-CRLF-003	NOTE	all	`--encoding`
OVERRIDE-MISSING-FILE	ERROR	settings	`--overrides`, `--siteaccess`, `--all-siteaccesses`
OVERRIDE-ENGINE-MISMATCH	WARN	settings	`--overrides`, `--siteaccess`, `--all-siteaccesses`
PARITY-PAIR	NOTE	all	`--parity`

Rule	Severity	Engine	Turned on by
ENGINE-UNAVAILABLE	ERROR	stpl, twig	an engine's library is missing
CHECK-READ	ERROR	all	a file cannot be read
CHECK-INTERNAL-ERROR	ERROR	all	the checker failed on a file

Severities: `ERROR` fails a run by default; `WARN` and `NOTE` fail it only with `--fail-on=warn` or `--fail-on=note`.

TPL-SYNTAX-001

ERROR, Exponential templates, every level. The template parser cannot read the `.tpl`: an end tag that does not match the open one, an end tag without a start tag, extra characters in an expression, a bad quote. The parser of the template system is used (as `eztemplatecheck` uses it), so the message is the one the site's debug output would show.

Bad:

```
{foreach $items as $item}
    {$item|wash}
{/if}
```

```
ERROR  tpl   var/tmp/check-docs/design/templates/broken.tpl:3
       Unterminated tag "foreach" does not match tag "if" [TPL-SYNTAX-001]
       > 3 | {/if}
```

Good: close what you open, `{/foreach}`. Another case, a closing tag too many:

```
ERROR  tpl   var/tmp/check-docs/rules2/w8.tpl:1:23
       End tag "foreach" without a start tag, ignoring tag [TPL-SYNTAX-001]
       > 1 | {foreach $l as $x}{/foreach}{/foreach}
```

An `{if}` left open at the very end of a file is closed by the parser without a message; the site behaves the same, so there is no finding.

TPL-SYNTAX-002

WARN, Exponential templates, every level. The template parser wrote a warning (not an error) to its log while reading the `.tpl`: something it guessed or left out. The template renders. Most warnings of the template system come when a template runs (an unknown function, a missing operator parameter), not while it is parsed, so this rule is rare; when it fires, the message is the parser's own.

TPL-COMPILE-001

ERROR, Exponential templates, `compile` and `strict`. The `.tpl` parses, but the template compiler cannot compile it, or the PHP it writes is not valid (the compiled file is checked by PHP's own parser, not run). The compiled file is written to a private directory below `var/cache/template-check/` and removed at the end; `--keep-compiled` keeps it so you can look at it. When template compilation is off (`site.ini [TemplateSettings]`

`TemplateCompile=disabled`), this step is left out.

Fix: change the construct of the template the message names; if the template is correct and the compiler is wrong, that is a bug to report with the kept compiled file.

TPL-STRICT-INCLUDE-MISSING

ERROR, Exponential templates, `strict` or `--includes`. An `{include uri='design:...'}` with a constant URI names a template that no design of the current siteaccess's chain has. Run with `-s <siteaccess>` to resolve in the chain of the siteaccess the template belongs to.

Bad:

```
{include uri='design:parts/no_such_part.tpl'}
```

```
ERROR  tpl   var/tmp/check-docs/rules/include.tpl:1
       the included template design:parts/no_such_part.tpl is not found in the design chain [TPL-
STRICT-INCLUDE-MISSING]
       > 1 | {include uri='design:parts/no_such_part.tpl'}
```

Good: the name of a template that exists in one of the designs (`design:parts/title.tpl`), or add the missing template.

TPL-STRICT-UNKNOWN-OPERATOR

WARN, Exponential templates, `strict`. An operator of the `.tpl` is not registered in the installation: when the template runs, the template system passes the operator's input through and writes a warning to the debug output. Usually a typing mistake or an operator of an extension that is not active. The hint names the closest registered operator.

Bad:

```
ok
{$name|wahs}
```

```
WARN   tpl   var/tmp/check-docs/rules/operator.tpl:2
       the operator wahs is not registered: it passes its input through [TPL-STRICT-UNKNOWN-OPERA
TOR]
       > 2 | {$name|wahs}
       hint: did you mean "wash"?
```

Good: `{$name|wash}`, or activate the extension that has the operator.

STPL-SYNTAX-001

ERROR, Smarty templates, every level. Smarty 5 cannot compile the `.stpl` with the bridge loaded (the template operators as modifiers, the `exp*` tags and blocks): an unclosed block, an unknown tag, a bad expression. Smarty has no separate parse step, so this rule covers the `syntax` level too.

Bad:

```
{if $node}
<h1>{$node.name|wash}</h1>
```

```
ERROR  stpl  var/tmp/check-docs/design/templates/broken.stpl:3
       unclosed {if} tag [STPL-SYNTAX-001]
```

Good: `{/if}` after the heading.

STPL-COMPILE-001

ERROR, Smarty templates, `compile` and `strict`. Smarty compiled the `.stpl`, but the PHP it wrote is not valid (checked by PHP's own parser in memory). Nothing is written to Smarty's compile directory.

STPL-SECURITY-001

ERROR, Smarty templates, every level, when `smarty.ini [SmartySettings] Security=enabled` (the default). Smarty's security policy, as the site applies it, refuses a construct of the `.stpl`: a tag, a modifier or a PHP function the policy does not allow. Because Smarty checks its policy while compiling, the refusal comes with the compile step.

Bad:

```
{eval var='x'}
```

Smarty refuses `{eval}`, `{fetch}` and the other tags of `smarty.ini SecurityDisabledTagList[]`, whether it words the refusal "not allowed by security setting" or "disabled by security setting":

```
ERROR  stpl  var/tmp/check-docs/rules/security.stpl:1
       tag 'eval' disabled by security setting [STPL-SECURITY-001]
       > 1 | {eval var='x'}
```

Good: the bridged form (`{expinclude}` for another template, `fetch()` as a template operator for content), or allow the construct in the settings when you trust it.

STPL-STRICT-INCLUDE-MISSING

ERROR, Smarty templates, `strict` or `--includes`. An `{expinclude uri='design:...'}` with a constant URI names a template that no design of the current siteaccess's chain has.

```
ERROR  stpl  var/tmp/check-docs/rules/include.stpl:1
       the included template design:parts/no_such_part.tpl is not found in the design chain [STPL
-STRICT-INCLUDE-MISSING]
       > 1 | {expinclude uri='design:parts/no_such_part.tpl'}
```

TWIG-SYNTAX-001

ERROR, Twig templates, every level. Twig 3 cannot tokenize or parse the `.twig` with the bridge, its tags and node visitors: an unclosed `{{` or `{%`, an unclosed block, an unknown tag, filter or function. For an unknown filter or function the hint names the closest template

operator.

Bad:

```
{% if node %}
<h1>{{ node.name|wash }</h1>
{% endif %}
```

```
ERROR  twig  var/tmp/check-docs/design/templates/broken.twig:2
       Unexpected "}". [TWIG-SYNTAX-001]
       > 2 | <h1>{{ node.name|wash }</h1>
```

Good: `{{ node.name|wash }}`.

TWIG-COMPILE-001

ERROR, Twig templates, `compile` and `strict`. Twig parsed the `.twig` but cannot compile it, or the compiled PHP is not valid (checked by PHP's own parser in memory). Nothing is written to Twig's cache directory.

TWIG-SECURITY-001

ERROR, Twig templates, `strict` or `--security`. The Twig sandbox, with the policy of `twig.ini` as the site applies it (`Security=enabled` by default), would refuse a tag, filter or function of the `.twig`. Every template operator, template function, block function and `exp*` tag is allowed; Twig's own tags, filters and functions only as `SecurityAllowedTagList[]`, `SecurityAllowedFilterList[]` and `SecurityAllowedFunctionList[]` list them.

Bad:

```
{% do 1 %}
```

```
ERROR  twig  var/tmp/check-docs/design/templates/parts/sandbox.twig:1
       Tag "do" is not allowed. (the sandbox would refuse the template) [TWIG-SECURITY-001]
       > 1 | {% do 1 %}
```

Good: `{% set %}` for a value, a template operator for a computation, or add the tag to `SecurityAllowedTagList[]` in a siteaccess whose templates you trust.

TWIG-STRICT-UNKNOWN-OPERATOR

WARN, Twig templates, `strict`. A filter or function named `exp_<name>` names no template operator. The converter writes `exp_<name>` for an operator whose name Twig has itself (`exp_count`, `exp_merge`), and for an operator the `.tpl` used but the installation does not have; such a name passes its input through, as an unknown operator of a `.tpl` does, and writes an error to the debug output when it runs. Usually a typing mistake or an extension that is not active.

Bad:

```
{{ node.name|exp_washh }}
```

```
WARN   twig  var/tmp/check-docs/design/templates/menu.twig:2
       exp_washh names no template operator: it passes its input through [TWIG-STRICT-UNKNOWN-OPE
RATOR]
       > 2 | {{ node.name|exp_washh }}
       hint: did you mean "wash"?
```

Good: `{{ node.name|wash }}`, or activate the extension that has the operator.

TWIG-STRICT-INCLUDE-MISSING

ERROR, Twig templates, `strict` or `--includes`. A `{% expinclude uri='design:...' %}` with a constant URI names a template that no design of the current siteaccess's chain has.

```
ERROR  twig  var/tmp/check-docs/design/templates/parts/missing.twig:1
       the included template design:parts/no_such_part.tpl is not found in the design chain [TWIG
-STRICT-INCLUDE-MISSING]
       > 1 | {% expinclude uri='design:parts/no_such_part.tpl' %}
```

TWIG-DEPRECATION-001

WARN, Twig templates, `--deprecations`. Twig raised a deprecation notice while parsing or compiling the `.twig`: a construct that still works but will change in a later Twig version. The message is Twig's own and names the replacement.

ENCODING-UTF8-001

ERROR, every engine, `--encoding`. The file is not valid UTF-8, for example saved in ISO-8859-1:

```
ERROR  tpl   var/tmp/check-docs/rules/latin1.tpl
       the file is not valid UTF-8 [ENCODING-UTF8-001]
```

Good: save the file as UTF-8.

ENCODING-BOM-002

WARN, every engine, `--encoding`. The file starts with a byte order mark. The engines print it, so it ends up in the page before the doctype or in the middle of an include.

```
WARN   tpl   var/tmp/check-docs/pair/bom.tpl:1:1
       the file starts with a byte order mark, which is printed [ENCODING-BOM-002]
```

Good: save the file as UTF-8 without a byte order mark.

ENCODING-CRLF-003

NOTE, every engine, `--encoding`. The file mixes CRLF and LF line endings. The engines print the line endings as they are, so a page can come out with both; usually the file was edited on two systems.

```
NOTE   tpl   var/tmp/check-docs/pair/bom.tpl
       the file mixes CRLF and LF line endings [ENCODING-CRLF-003]
```

OVERRIDE-MISSING-FILE

ERROR, settings, with `--overrides`, `--siteaccess` or `--all-siteaccesses`. An override rule of a siteaccess (`override.ini [<rule>] MatchFile=`) points to a file that no design of the siteaccess's chain has below `override/<Subdir>/`. The finding is on the siteaccess's `settings/siteaccess/<siteaccess>/override.ini.append.php`, and its message has the form `[<rule>] MatchFile=<file>: no design of the siteaccess <siteaccess> has it`.

Good: correct `MatchFile`, or add the template to a design of the chain.

OVERRIDE-ENGINE-MISMATCH

WARN, settings, with `--overrides`, `--siteaccess` or `--all-siteaccesses`. An override rule points to a file that exists but whose ending is not a template engine's (`full/article.html` instead of `full/article.twig`), so the template system would read it as a `.tpl`.

PARITY-PAIR

NOTE, every engine, `--parity`. The same template exists in two or more engines side by side, for example `title.tpl` and `title.twig` in one directory. Useful after a conversion to see which templates have a counterpart; which file the site uses is decided by the override rules.

```
NOTE   tpl   var/tmp/check-docs/pair/title.tpl
       the same template in 2 engines: var/tmp/check-docs/pair/title.tpl, var/tmp/check-docs/pair
/title.twig [PARITY-PAIR]
```

`--parity=render` (experimental) reports the same and says that the rendering comparison is not done yet.

ENGINE-UNAVAILABLE

ERROR, Smarty and Twig templates. The engine of the file is not available: its library (Smarty 5 or Twig 3) is not installed, or the checker registered for it in `site.ini [TemplateBridgeSettings] CheckerList[]` does not exist. The finding comes once per engine, on the first of its files; the files are skipped (`SKIP`), the console prints `UNAVAILABLE <engine>`, and the exit code is 3 unless `--allow-missing-engine` is given. This finding never counts for `--fail-on`. `./console exp:template:engines` shows which engines are available.

CHECK-READ

ERROR, every engine. The file cannot be read (permissions, a broken link).

CHECK-INTERNAL-ERROR

ERROR, every engine. The checker of the engine failed on the file with an exception; the message names it. The other files are still checked and the exit code is 1, not 4 (4 is for a failure of the command itself). Report it with the file.

The machine output

`--format=json` writes one document of the schema `exp-template-check/1`, described as a JSON Schema in template-check-schema.json:

Field	Contents
`schema`	always `exp-template-check/1`
`summary`	`files`, ok (files without findings), `skipped`, `errors`, `warnings`, `notes` (finding counts), `engines` (files per engine), `unavailable` (engines), `seconds`, `level`, `exit` (the exit code), `result` (PASS or FAIL)
`findings`	one object per finding: `rule`, `severity`, `engine`, `file`, `line` and `column` (0 when not known), `message`, `hint` (or null), `source` (the source line, or null), `belongs` (where the file belongs: designs, extensions, siteaccesses)
`files`	one object per file checked: `file`, `engine`, `status` (`OK`, `ERROR`, `WARN`, `NOTE`, `SKIP`), `ms`, `belongs`

`--format=jsonl` writes the same findings one per line with `"type": "finding"`, and the summary as the last line with `"type": "summary"`; each line also carries `schema`. Fields may be added in later releases, never removed or changed in type.

Exponential Copyright © 1998-2026 Exponential Foundation and 7x
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".
This documentation is free documentation, licensed under the GNU Free Documentation License 1.3 or later:
https://www.gnu.org/licenses/fdl-1.3.html.