

# Customizing Notifications

TeamCity provides a wide range of notification possibilities to keep developers informed about the status of their projects. Notifications can be sent by e-mail, Jabber/XMPP instant messages or can be displayed in the IDE (with the help of TeamCity plugins) or the Windows system tray (using TeamCity Windows tray notifier). Each user can [select the events to receive notifications](#). The notification messages can be customized globally on per-server basis.

Notifications can also be received via Atom/RSS syndication feeds, but since feed use "pull" model for receiving notifications instead of "push", some of the approaches are different for the feeds.

## Notifications Lifecycle

TeamCity supports a set of events that can generate user notifications (such as build failure, investigation state changes, etc). On event occurrence, for each notifiator type, TeamCity processes notification settings for all the users to define users that the notification should be sent to.

When the set of users is determined, TeamCity fills the notification model (the objects relevant to the notification as "build", investigation data, etc.) and evaluates a notification template that corresponds to the notification event.

The template uses the data model objects to generate output values (e.g. notification message text). The output values are then used by the notifiator to send the message. Each notifiator supports a specific set of the output values.

Please note that the template is evaluated once for an event which means that notification properties cannot be adjusted on per-user basis.

The output values defined by the template are then used by the notifiator to send notification to the selected users.

## Customizing Notifications Templates

Notification Templates Location

Each of the bundled notifiators has a directory under `<TeamCity data directory>/config/_notifications/` which stores [FreeMarker](#) (.ftl) templates. There are also `.dist` files that store default templates. Each notification type evaluates a template file with corresponding name. The template files can be modified while the server is running. By default server checks for changes in the files each 60 seconds, but this can be changed by setting `teamcity.notification.template.update.interval` [internal property](#) to the desired number of seconds.

If there an error occurs during template evaluation, TeamCity logs the error details into `teamcity-notifications.log`. There can be non-critical errors that result in ignoring part of the template or critical errors that result in inability to send notification at all. Whenever you make changes to the notification templates please ensure the notification can still be sent.

This document doesn't describe the FreeMarker template language, so if you need a guidance on the FreeMarker syntax, please refer to the corresponding template manual at <http://freemarker.org/docs/dgui.html>.

TeamCity notifiators use templates to evaluate output values (global template variables) which are then retrieved by name. The following output values are supported:

Email Notifiator

- `subject` - subject of the email message to send
- `body` - plain text of the email message to send
- `bodyHtml` - (optional) HTML text of the email message to send. It will be included together with plain text part of the message
- `headers` - (optional) Raw list of additional headers to include into email. One header per line. For example:

```
<#global headers>
  X-Priority: 1 (Highest)
  Importance: High
</#global>
```

Jabber

- `message` - plain text of the message to send

IDE Notifications and Windows Tray Notifications

- message - plain text of the message to send
- link - URL of the TeamCity page that contains detailed information about the event

The Atom/RSS feeds template differs from the others. For the details, please refer to the [dedicated section](#).

For the template evaluation TeamCity provides the default data model that can be used inside the template. The objects exposed in the model are instances of the corresponding classes from [TeamCity server-side open API](#).

The set of available objects model differs for different events.

You can also add your own objects into the model via plugin. See [Extending Notification Templates Model](#) for details.

Here is an example description of model (the code can be used in IntelliJ IDEA to edit the template with completion):

```
<!-- @ftlvariable name="project" type="jetbrains.buildServer.serverSide.SProject" -->
<!-- @ftlvariable name="buildType" type="jetbrains.buildServer.serverSide.SBuildType" -->
<!-- @ftlvariable name="build" type="jetbrains.buildServer.serverSide.SBuild" -->
<!-- @ftlvariable name="agentName" type="java.lang.String" -->
<!-- @ftlvariable name="buildServer" type="jetbrains.buildServer.serverSide.SBuildServer" -->
<!-- @ftlvariable name="webLinks" type="jetbrains.buildServer.serverSide.WebLinks" -->

<!-- @ftlvariable name="var.buildFailedTestsErrors" type="java.lang.String" -->
<!-- @ftlvariable name="var.buildShortStatusDescription" type="java.lang.String" -->
<!-- @ftlvariable name="var.buildChanges" type="java.lang.String" -->
<!-- @ftlvariable name="var.buildCompilationErrors" type="java.lang.String" -->

<!-- @ftlvariable name="link.editNotificationsLink" type="java.lang.String" -->
<!-- @ftlvariable name="link.buildResultsLink" type="java.lang.String" -->
<!-- @ftlvariable name="link.buildChangesLink" type="java.lang.String" -->
<!-- @ftlvariable name="responsibility" type="jetbrains.buildServer.responsibility.ResponsibilityEntry"
-->
<!-- @ftlvariable name="oldResponsibility"
type="jetbrains.buildServer.responsibility.ResponsibilityEntry" -->
```

#### TeamCity notification properties

The following [properties](#) can be useful to customize the notifications behaviour:

teamcity.notification.template.update.interval - how often the templates are reread by system (integer, in seconds, default 60)

teamcity.notification.includeDebugInfo - include debug information into the message in case of error (boolean, default false)

teamcity.notification.maxChangesNum - max number of changes to list in e-mail message (integer, default 10)

teamcity.notification.maxCompilationDataSize - max size (in bytes) of compilation error data to include in e-mail message (integer, default 20480)

teamcity.notification.maxFailedTestNum - max number of failed tests to list in e-mail message (integer, default 50)

teamcity.notification.maxFailedTestStacktraces - max number of test stacktraces in e-mail message (integer, default 5)

teamcity.notification.maxFailedTestDataSize - max size (in bytes) of failed test output data to include in a single e-mail message (integer, default 10240)

## Syndication Feed Template

The template uses different approach to configuration from other notification engines.

The default template is stored in the file: `<TeamCity data directory>/config/default-feed-item-template.ftl`. This file should never be edited: it is overwritten on every server startup with the default copy. To specify a new template to use, copy the file under the name `feed-item-template.ftl` into the same directory. This file can be edited and will not be overwritten. It will be used by the engine if present.

The template is a [FreeMarker](#) template and can be freely edited.

You can use several templates on the single sever. The template name can be passed as a [URL parameter](#) of the feed URL.

During feed rendering, the template is evaluated to get the feed content. The resultant content is defined by the global variables defined in the temple.

See the default template for an example of available input variables and output variables.

See also:

[User's Guide: Subscribing to Notifications](#)