Shared Resources

The Shared Resources build feature allows limiting concurrently running builds using a shared resource, such as an external (to the CI server) resource, e.g. test database, or a server with a limited number of connections, etc.

Some of such resources may be accessed concurrently but allow only a limited number of connections, others require exclusive access. Adding different locks to shared resources addresses these cases: now you can define a resource on the project level, configure its parameters (e.g. type and quota) and then use this resource in specific build configurations by adding the Shared Resources build feature to them. The build starts once the lock on the resource is acquired; on the build completion the lock is released. While the builds using the resource are running, the resource is unavailable, and the other builds requiring locks will be waiting in the queue.

On this page:

Adding and Editing Shared Resources

You can add, edit shared resources, and explore their details (origin of the resource, its usage, etc.) on the Shared Resources t ab of the project configuration page.

The resource name can contain only alpha-numeric characters and underscores ("_") - maximum 80 characters - and should start with a Latin letter.

Shared resources defined at a project level are available in all its subprojects and build configurations.

On the subproject level, it is not possible to edit a resource inherited from a project.

However, it is possible to redefine a resource inherited from a project by creating a resource with the same name but with different settings in a subproject.
For example:

1. Create a resource A with the infinitive quota in project Parent.
2. Go to its subproject Child 1, and created a resource A (the same name as the resource in Parent) with the quota of 5. TeamCity will treat this resource as redefined in subproject Child 1: i.e. the settings of the resource A defined at the project level (infinite quota) will be propagated to all the other subprojects, with the exception of subproject Child 1 where resource A will have the quota of 5.

The usage of a resource is calculated by scanning the subtree of the current project; thus, if several projects use the same resource, only the usages in the current one will be counted.

Types of Shared Resources

When you click Add new resource, three types of resources are available:

- Infinite resource is a shared resource with an unlimited number of read locks.
- Resource with quota: quota is a maximum number of read locks that can be acquired on the resource.
- Resource with custom values: a custom value (e.g. a URL) is passed to the build that has acquired a lock on such a resource.

Using Shared Resources in Build Configurations

To define which build configuration(s) will use the resources, add this build feature to the build configuration settings.

When adding a shared resource, you need to select a resource available to this build configuration and define a lock.

The following locks are available:

Locks for Resources with Quotas

For this resource type, two types of locks are supported:

- Read locks - shared (multiple running builds with read locks are allowed).
- Write locks - exclusive (only a single running build with a write lock is allowed).

A resource with a quota will allow concurrent access to multiple builds for reading but will restrict access to a single build for writes to the resource. Until all read locks are released, the build requiring a write lock will not start and will wait in the queue while new readers are able to acquire the lock.
⚠ A build with a resource quota set to zero will not run.

Lock Fairness

While read locks enable multiple builds to concurrently access a shared resource with a quota > 1, lock fairness ensures that the build queue is not violated. It means, that if there is shared resource with a quota >1, and there are several queued builds with read locks and a build with a write lock in between the readers, the build with the write lock will wait until all builds with read locks earlier in the queue finish and release the lock. Then the build requiring a write lock will be executed, and only after that the other readers can acquire the lock. Lock fairness does not allow builds with read locks interfere with build queue ordering and 'slip' past the build that is waiting for a write lock to become available.

Locks for Resources with Custom Values

Resources with custom values support three types of locks:

- Locks on any available value: a build that uses the resource will start if at least one of the values is available. If all values are being used at the moment, the build will wait in the queue.
- Locks on all values: a build will lock all the values of the resource. No other builds that use this resource will start until the current one is finished.
- Locks on specific value: only a specific value of the resource will be passed to the build. If the value is already taken by a running build, the new build will wait in the queue until the value becomes available.

When the resource is defined and the locks are added, the build gets a configuration parameter with the name of the lock and with the value of the resource string (`teamcity.locks.readLock.<lockName>` or `teamcity.locks.writeLock.<lockName>`), e.g. the parameter name can be: `teamcity.locks.readLock.databaseUrl`.

⚠ Locks for resources with custom values cannot be added to composite builds: TW-54381

Locks for resources in composite builds

Shared resources (except ones with custom values) can now be locked for composite builds as well. A lock on the specified resource will be acquired when a composite build starts (when the first build of the composite build chain starts); the lock will be released when the composite build finishes (the last build in its chain is finished).

The locks acquired on composite builds affect only these composite builds and are not propagated to their individual parts. For example, if a resource has a quota of N, then N composite builds that have a read lock on this resource can be run concurrently. The number of concurrent individual builds inside these composite builds will not be affected by the resource quota.

Viewing Shared Resources Usage

Since TeamCity 2018.1, if at least one resource lock is defined in a build configuration, you can view the resources used by the build on the Shared Resources tab of the Build Results page. The tab displays the resources and their type, including the locks used by the build for each resource.

Clicking the resource name takes you back to the the shared resources configuration page on the project level.

Development Links

See the Shared Resources plugin page

See also:

- JetBrains TV TeamCity Shared Resources Screencast