Working with Meta-Runner

A Meta-Runner allows you to extract build steps, requirements and parameters from a build configuration and create a build runner out of them. This build runner can then be used as any other build runner in a build step of any other build configuration or template.

Basically, a meta-runner is a set of build steps from one build configuration that you can reuse in another; it is an xml definition containing build steps, requirements and parameters that you can utilize in xml definitions of other build configurations. TeamCity allows extracting meta-runners using the web UI.

With a meta-runner, you can easily reuse existing runners, create new runners for typical tasks (e.g. publish to FTP, delete directory, etc.), you can simplify your build configuration and decrease a number of build steps.

All meta-runners are stored on a project level, so they are available within this project and its subprojects only, and are not visible outside. If a meta-runner is stored on the <Root project> level, it is available globally (in all projects).

You can use the existing meta-runners from the TeamCity Meta-Runners Power Pack or create your own meta-runner.

On this page:

- Using Meta-Runners Power Pack
  - Installing Meta-Runner
- Creating Meta-Runner
  - Preparing Build Configuration
  - Verifying Build Configuration Works Properly
  - Extracting and Using Meta-Runner
- Creating Meta-Runner from XML Definition of Build Configuration
- Creating Build Configuration from Meta-Runner

Using Meta-Runners Power Pack

Meta-runners Power Pack for TeamCity available on GitHub is a collection of meta-runners for various tasks like downloading a file, triggering a build, tagging a build, changing a build status, running PHP tasks, etc.

Each file called *MRPP_\<some text\>.xml* contains a definition of a single Meta-runner. Download the required meta-runner (or copy its definition to a file) and install it as described in the section below.

Installing Meta-Runner

Since TeamCity 9.0, you can install a meta-runner using the TeamCity Web UI. Alternatively, you can do it directly via the file system.

- to install a Meta-runner via the Web UI, go to the Project Settings page, select Meta-Runners from the list of settings on the left, click Upload Meta-Runner and select the Meta-runner definition file. Save you changes.

- to install a Meta-runner directly to the file system, take the Meta-runner definition file and put it into the `TeamCity Data Directory\config\projects\<project ID>\pluginData\metaRunners` directory, where `\<Project ID\>` is the identifier of a project where you want to place the Meta-runner. If the `metaRunners` directory does not exist, it should be created. Once you place the file on the disk, TeamCity will detect it and load this Meta-runner; no server restart is required.

Creating Meta-Runner

Let us consider an example of creating a meta-runner.

To create a meta-runner, follow these steps (described below in more detail):

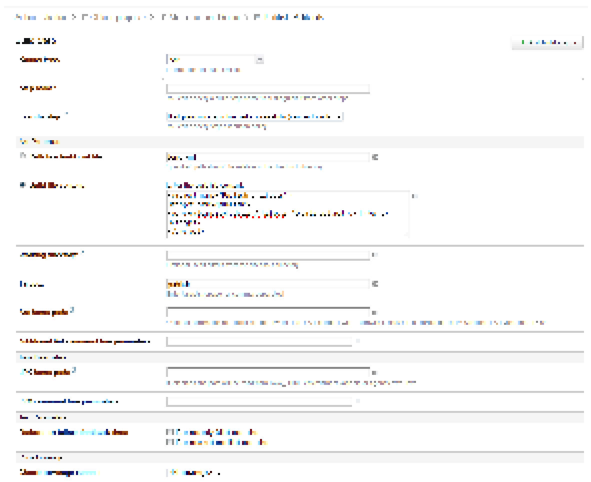
1. prepare a build configuration to test the build steps we are going to use in our meta-runner,
2. make sure the build configuration is working,
3. extract a meta-runner to the desired project.

In this example, we will create a meta-runner to publish some artifacts to TeamCity with the help of corresponding service message.

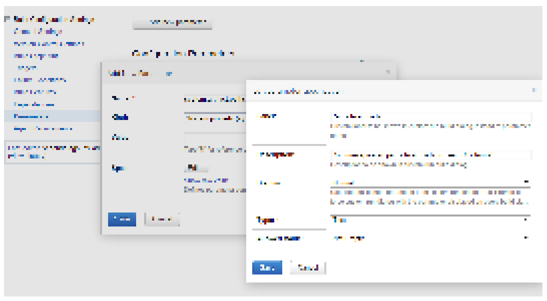
Usually artifacts configured in a build configuration are published when the build finishes. However, sometimes for long builds with multiple build steps we need artifacts faster. In this example, we will create a runner which can be inserted between any build steps and can be configured to publish artifacts produced by previous steps.

Preparing Build Configuration

The first step is to prepare a build configuration which will work the same way as the meta-runner we would like to produce. Let us use the configuration with a single Ant build step: Ant can be executed on any platform where the TeamCity agent runs; besides, Ant runner in TeamCity supports build.xml specified right in the runner settings. This is important because our build configuration must be self-contained, it cannot take build.xml from the version control repository. So in our case the Ant step settings will look like this:



where `artifact.paths` is a system property. We need to add it on the Parameters tab of the build configuration settings:

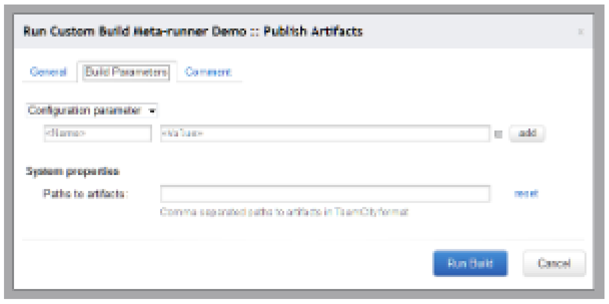


Note that each parameter can have a specification where we can provide the label, description, type of control and specify validation conditions. Before version 8.0 this specification was used by the custom build dialog only. Now this specification is used by a meta-runner too.

Here the Ant build step is used just as an example. In the initial build configuration, you can use any of the available build runners (e.g. MSBuild, .Net process, etc.) and configure the settings, define the parameters for this build step. When you extract a meta-runner from this build configuration, all the settings defined in the build step, and all the build parameters will be added to the meta-runner.

Verifying Build Configuration Works Properly

Once the build steps and parameters are defined, we need to make sure our build configuration works by running a couple of builds through the custom build dialog:



Run Custom Build Meta-runner Demo :: Publish Artifacts

General | Build Parameters | Comment

Configuration parameter

<Name> <Value> add

System properties

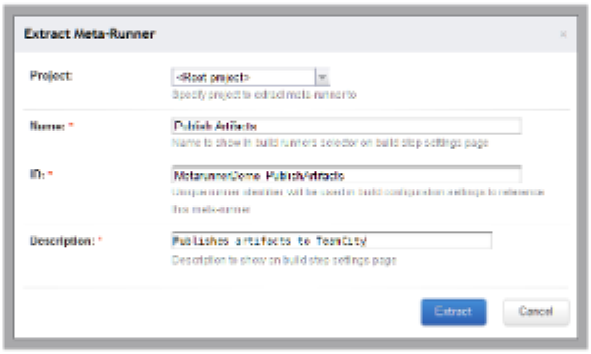
Paths to artifacts: reset

Comma separated paths to artifacts in TeamCity format

Run Build | Cancel

Extracting and Using Meta-Runner

If the build configuration works properly, we can create a meta-runner using the Actions button in the top right-hand corner, E xtract meta-runner... option:



Extract Meta-Runner

Project:	<Root project> Specify project to extract meta-runner to
Name: *	Publish Artifacts Name to show in build runners selector on build step settings page
ID: *	MetarunnerDemo_PublishArtifacts Unique runner identifier. Will be used in build configuration settings to reference this meta-runner
Description: *	Publishes artifacts to TeamCity Description to show on build step settings page

Extract Cancel

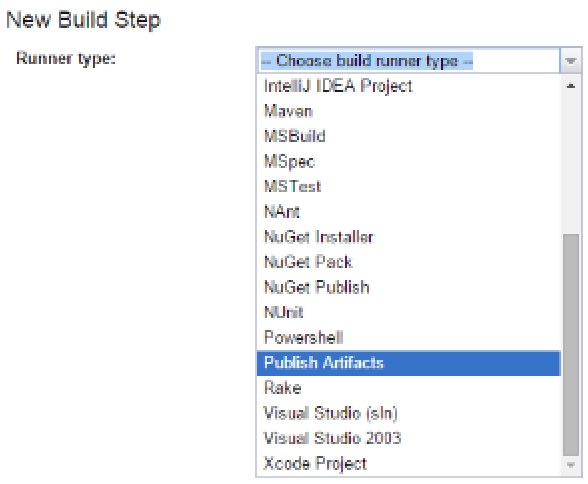
The Extract Meta-Runner dialog requires specifying the project where the meta-runner will be created. A meta-runner created in a project will be available in this project and all its subprojects. In our case the <Root project> is selected, so the meta-runner will be available in all projects.

We also need to provide the name, description and an ID for the meta-runner: the name and description will be shown in the web interface, an ID is required to distinguish this meta-runner from others.

Upon clicking the Extract button, TeamCity will take definitions of all build steps and parameters in this build configuration and create a build runner out of them.

Besides build steps and parameters, a meta-runner can also have requirements: if requirements are defined in the build configuration, they will be extracted to the meta-runner automatically. Requirements can be useful if the tools used by meta-runner are available on specific platforms only.

Once the meta-runner is extracted, it becomes available in the build runners selector and can be used in any build step just like any other build runner:

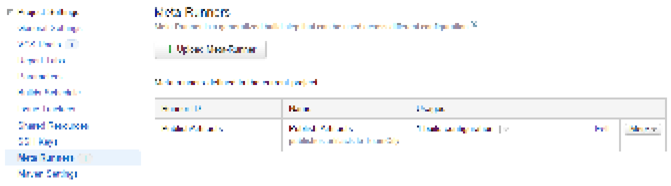


New Build Step

Runner type: -- Choose build runner type --

- IntelliJ IDEA Project
- Maven
- MSBuild
- MSpec
- MSTest
- NAnt
- NuGet Installer
- NuGet Pack
- NuGet Publish
- NUnit
- Powershell
- Publish Artifacts
- Rake
- Visual Studio (sln)
- Visual Studio 2003
- Xcode Project

The current meta-runner usages can be seen at the project Meta-Runners page:



Meta Runners

Upload Meta-Runner

Shared Resources

SSH Keys

Meta-Runners

When a meta-runner is extracted, all steps will be extracted. If you need to reorder parameters or make some quick fixes in the runner script, you can edit its raw xml definition in the web browser: go to the Administration page of the project -> Meta-Runners and use the Edit option next to the meta-runner. The parameters will be shown in the same order as the <param>

elements in the xml definition. Definitions of meta-runners are stored in the `TeamCity Data Directory` `\config\projects\<project ID>\pluginData\metaRunners` folder.

Creating Meta-Runner from XML Definition of Build Configuration

Alternatively, you can use the xml definition of an existing build configuration as a meta-runner. To do it, save the definition of this build configuration to a file named as follows: <runner id>.xml, where <runner id> is the ID of this build runner. Install the meta-runner as described above.

Since a meta-runner looks and works like any other runner, it is also possible to create another meta-runner on the basis of an existing meta-runner.

Creating Build Configuration from Meta-Runner

Since TeamCity 9.0, if you need to fix a meta-runner and test your fix, you can create a build configuration from a meta-runner, can change its steps, adjust parameters and requirements, check how it works, and then use the Extract meta-runner action to apply the changes to the existing meta-runner with the same ID.