

TeamCity Integration Guide

Integrating CodeSign Secure with TeamCity automates the code signing step within your build configurations, ensuring that your software artifacts are securely signed before they are released or deployed, without manual intervention.

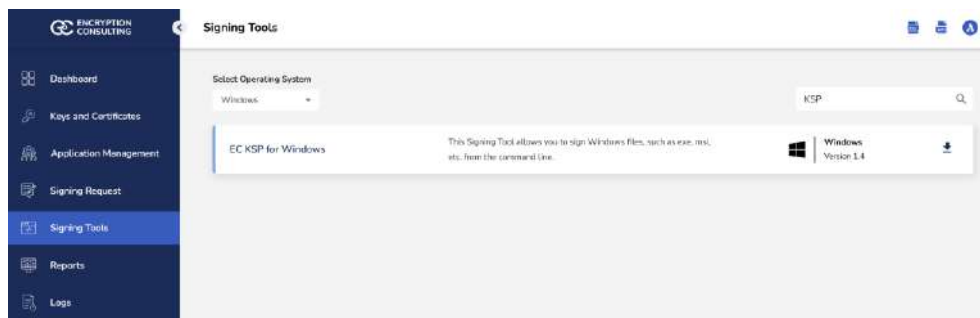
1. Set up CodeSign Secure KSP

The Encryption Consulting Key Storage Provider (KSP) for Windows is a software component that extends the Microsoft Cryptography API: Next Generation (CNG) framework. Its primary purpose is to enable Windows applications, such as signtool.exe, to interact seamlessly with the cryptographic keys and certificates stored within an HSM.

Steps:

Step 1: Download the EC KSP

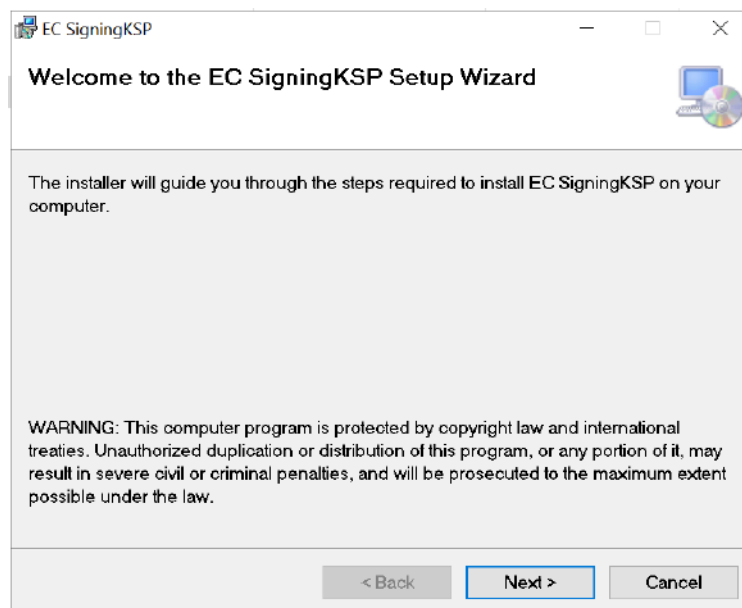
- Log in to the CodeSign Secure portal and navigate to the Signing Tools section to download “EC KSP for Windows.”



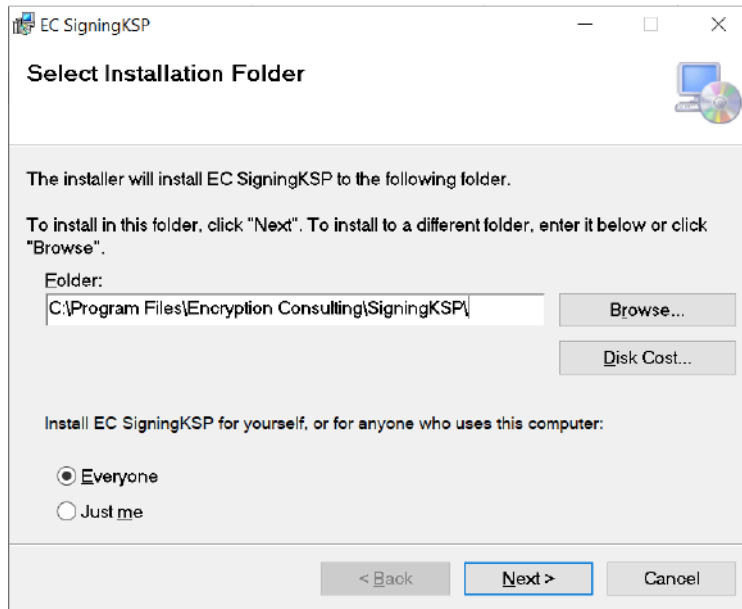
- Extract the zip file to get the “Setup.msi” file.

Step 2: Install the EC KSP

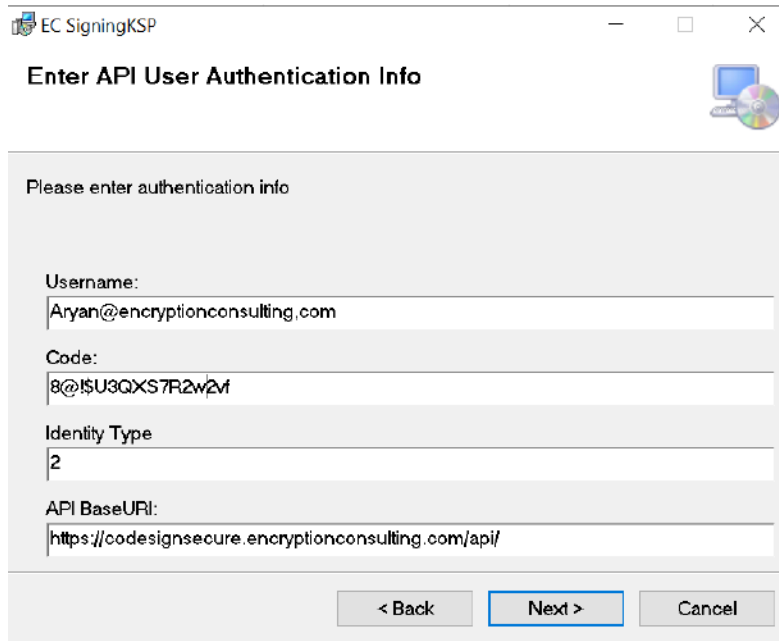
- Run the “Setup.msi” installer with Administrator privileges.



- Follow the on-screen prompts of the installation wizard.
 - Accept the End-User License Agreement.
 - Choose the installation directory (the default is C:\Program Files\Encryption Consulting\SigningKSP).
 - Choose whether you want to install the KSP for Everyone or just for the current user



- Enter the prompted details such as:
 - Username – The username/email that you use to log in to the CodeSign Secure portal
 - Code – The secret code that you set at the time of setting up the CodeSign Secure solution.
 - IdentityType – Keep this field as default (2)
 - CodeSign Secure URL – The URL to access the portal (Remember to add “/api/” at the end of the URL)



EC SigningKSP

Enter API User Authentication Info

Please enter authentication info

Username:
Aryan@encryptionconsulting.com

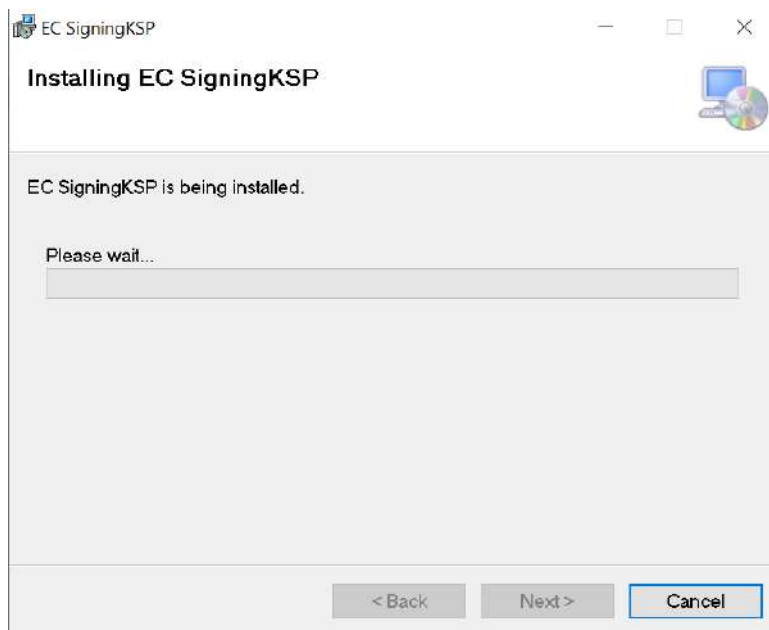
Code:
8@!\$U3QXS7R2w2M

Identity Type
2

API BaseURL:
https://codesignsecure.encryptionconsulting.com/api/

< Back Next > Cancel

- Click Next and confirm the installation.



EC SigningKSP

Installing EC SigningKSP

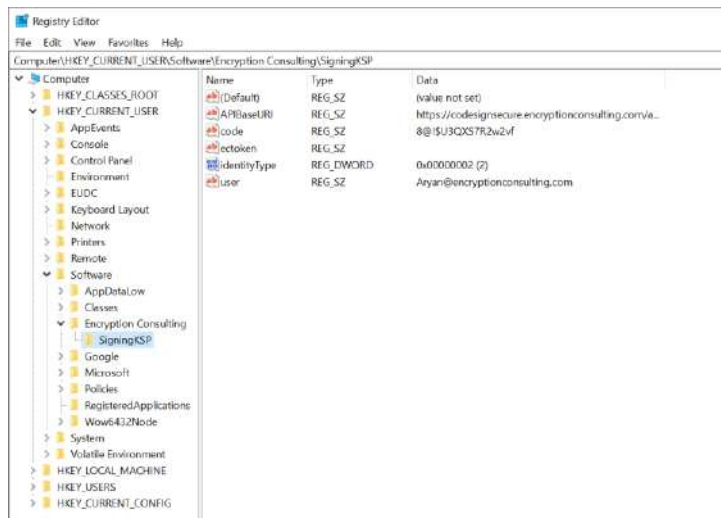
EC SigningKSP is being installed.

Please wait...

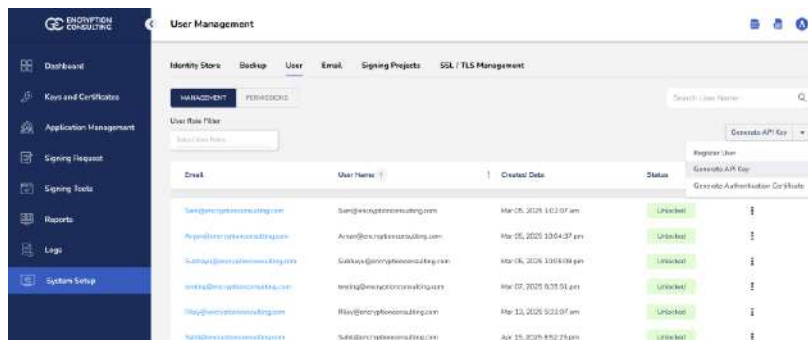
< Back Next > Cancel

Step 3: Configure the Registry Editor settings

- Open the Registry Editor and navigate to HKEY_CURRENT_USER>Software>Encryption Consulting>SigningKSP directory.



- Now open the CodeSign Secure portal and navigate to System Setup>User. Select the drop-down from the right side to “Generate API Key”.



- Create a token for your account by providing a name and the validity period. Remember to copy the token as it will be shown only once.

Generate API Key

API Key Name:

Expiry Date:

- Add this token to the “ectoken” field in the Registry Editor.

Name	Type	Data
(Default)	REG_SZ	(value not set)
APIBaseURL	REG_SZ	https://codesignsecure.encryptionconsulting.com/a...
code	REG_SZ	8@!\$U3QXS7R2w2vf
ectoken	REG_SZ	
identityType	REG_DWORD	0x00000002 (2)
user	REG_SZ	Aryan@encryptionconsulting.com

Edit String

Value name:
ectoken

Value data:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1c2VybmFtZSI6IkFyeWVfuQc

OK Cancel

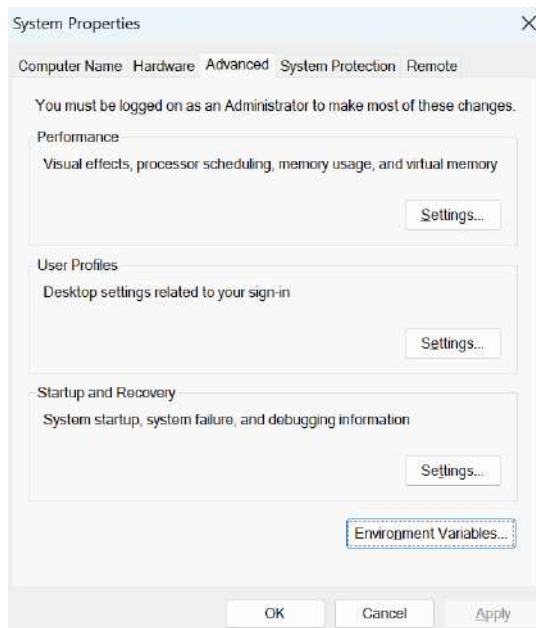
2. Set up P12 Authentication Certificate

Setting up a P12 Certificate involves configuring your environment variables to authenticate your client machine with Encryption Consulting's CodeSign Secure.

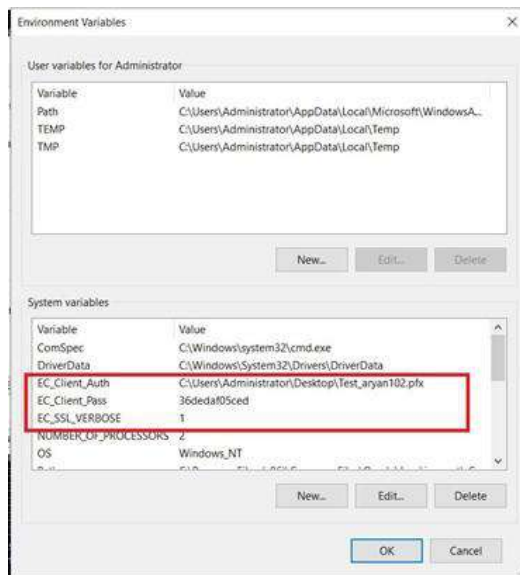
Steps:

Step 1: Configure the Environment Variables

- Open the Environment Variables from your Start Menu



- Add new system variables by clicking on the New button. Provide the following variable name and its corresponding details.
 - **EC_Client_Auth:** Corresponds to the path of your SSL Authentication certificate, which can be created from CodeSignSecure
 - **EC_Client_Pass:** Corresponds to the password of your certificate, which is provided at the time of creation of the certificate.
 - **EC_SSL_VERBOSE:** Corresponds to the setting to either enable (1) or disable (0) the debugging output for EC KSP.



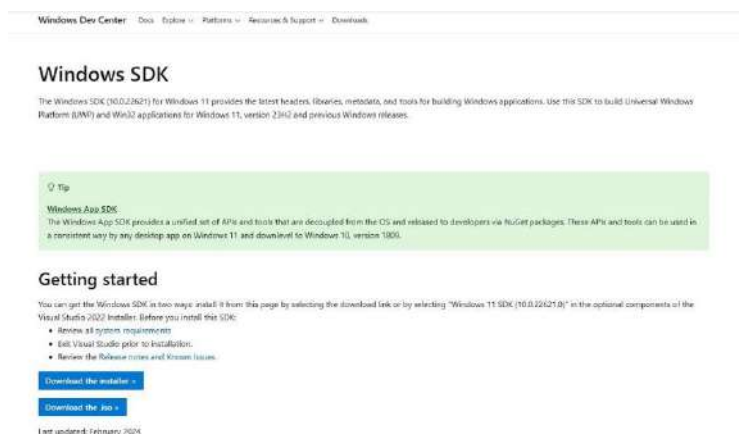
3. Set up Signtool for Signing

Setting up signtool for code signing involves ensuring that the Signtool.exe utility is available on your machine and configured to correctly interact with Encryption Consulting's cryptographic provider that provides access to your code signing certificate's private key.

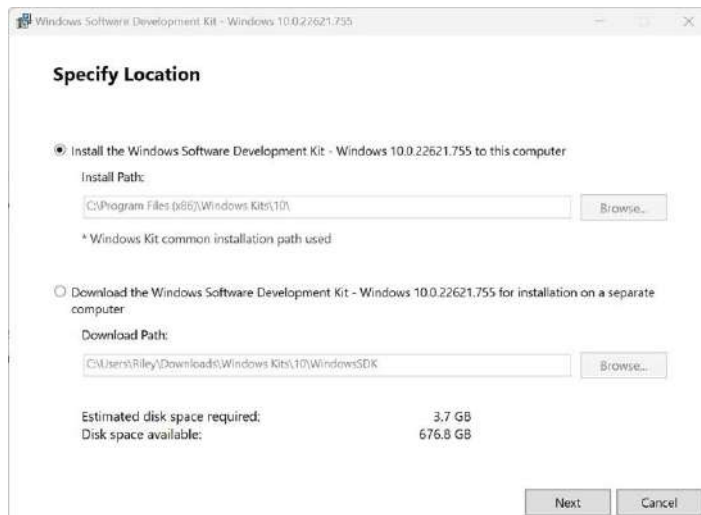
Steps:

Step 1: Download and Install Windows SDK

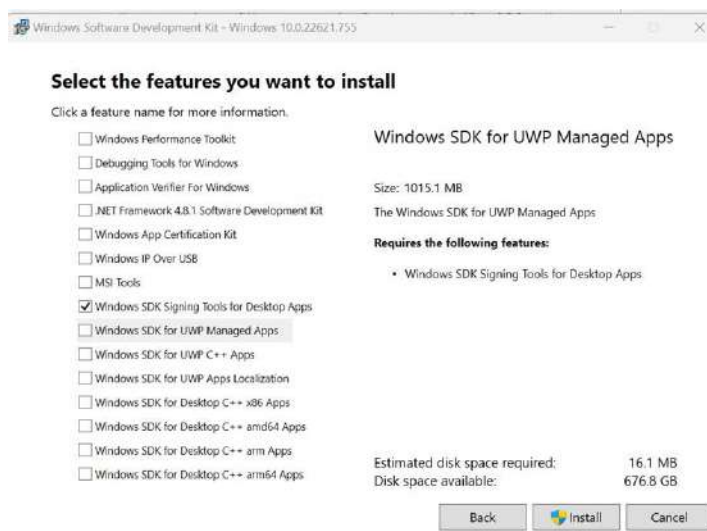
- Using the following download link, download the Windows Software Development Kit with the following tools selected: developer.microsoft.com/en-us/windows/downloads/windows-10-sdk/



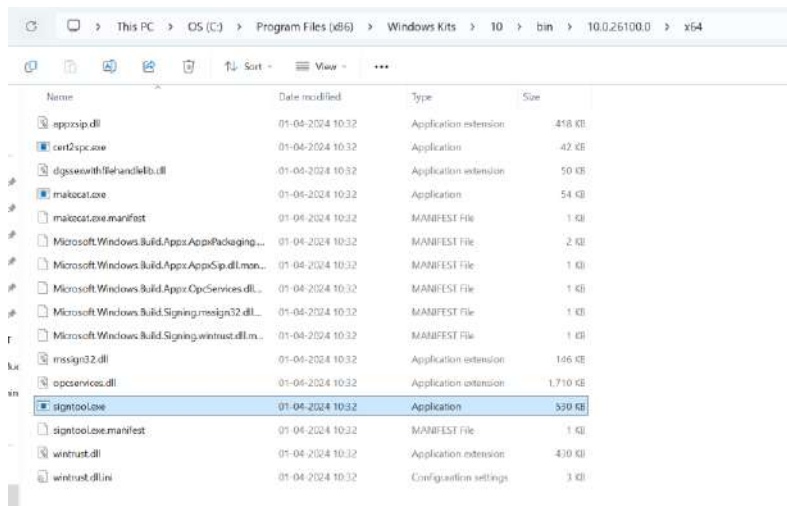
- Open the installer once downloaded and select “Next” on the first screen to keep the default settings.



- Follow the on-screen prompts of the installation wizard.
 - Accept the Windows Kit Privacy
 - Accept the End-User License Agreement.
- Deselect everything except "Windows SDK Signing Tools for Desktop Apps" and select "Install".



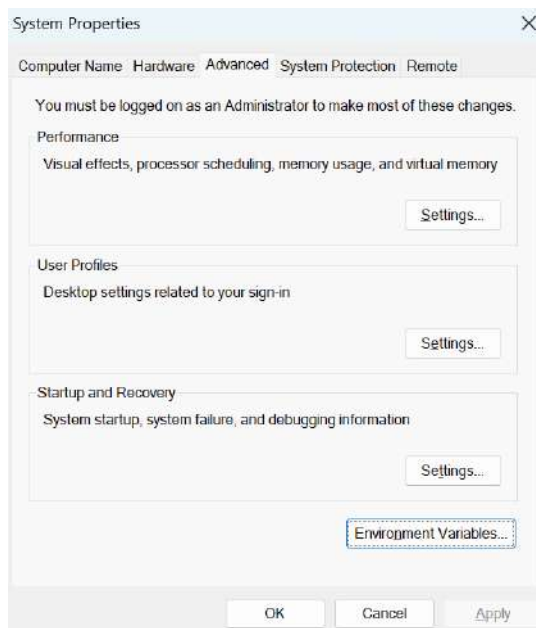
- Go to the following path where the tools should have been downloaded to: "C:\Program Files (x86)\Windows Kits\10\bin". Select the desired version directory and check whether the "signtool.exe" file is present.



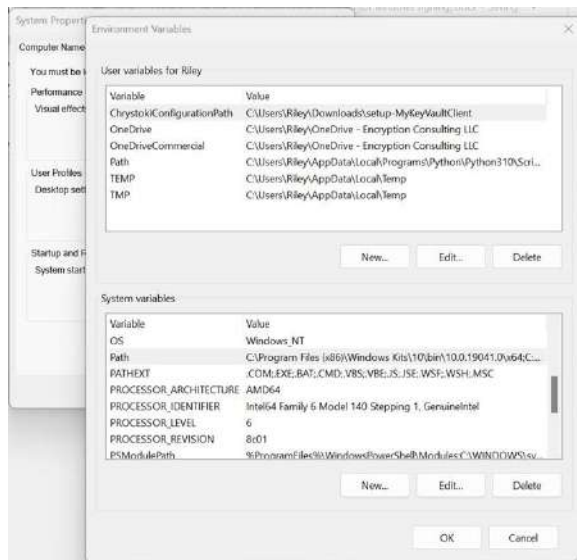
- Ensure you are in the x64 directory and copy this directory path.

Step 2: Add Path to Signtool.exe in Environment Variables

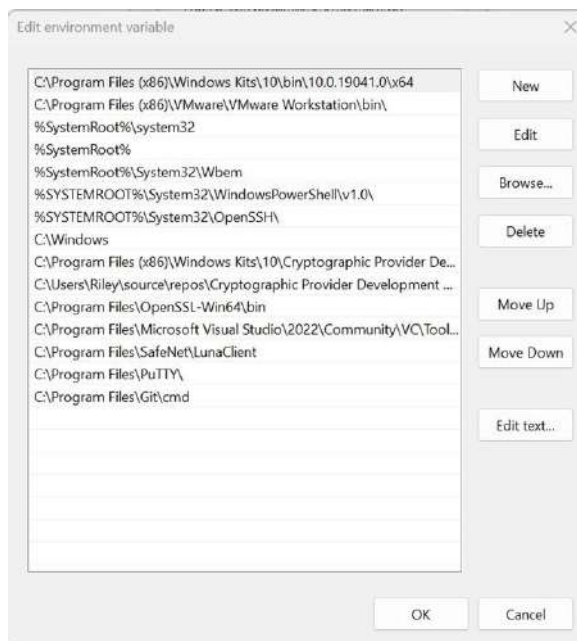
- Open the Environment Variables from the Start Menu.



- Scroll down through the system variables on the bottom table until you find PATH in the variable names.



- Double click on PATH in system variables and select New on the left of the screen. Paste your copied directory path of “signtool.exe” into the new selection.



- Select OK at the bottom to exit the Environment Variables page.

4. Set up TeamCity

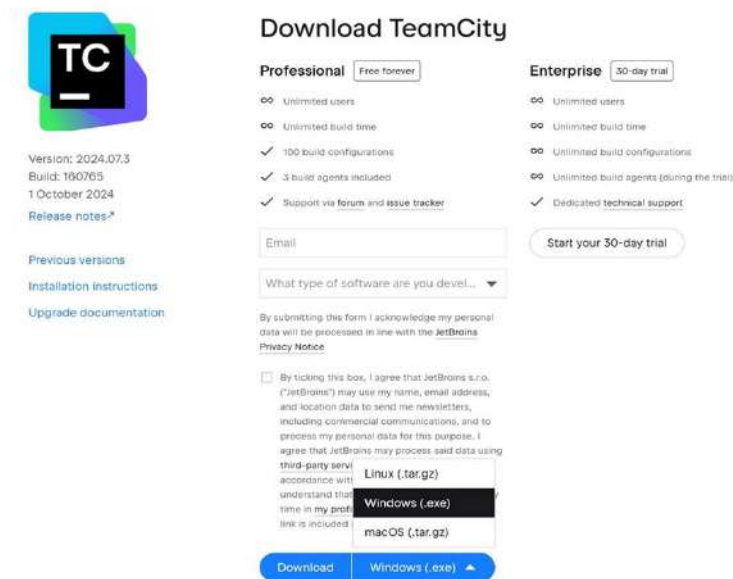
Setting up TeamCity involves preparing your TeamCity environment, specifically its build agents and build configurations, to perform automated code signing using Encryption Consulting's CodeSign Secure and KSP.

Steps:

Step 1: Download and Install the TeamCity Application

- You can either download the free TeamCity application or a paid Enterprise version using the [link](#). There are three OS choices for the

application: Linux, Windows, and macOS. We'll be working on the Windows executable file.



Download TeamCity

Professional Free forever

- Unlimited users
- Unlimited build time
- 100 build configurations
- 3 build agents included
- Support via forum and issue tracker

Enterprise 30-day trial

- Unlimited users
- Unlimited build time
- Unlimited build configurations
- Unlimited build agents (during the trial)
- Dedicated technical support

Email:

What type of software are you developing?

By submitting this form I acknowledge my personal data will be processed in line with the [JetBrains Privacy Notice](#)

☐ By ticking this box, I agree that JetBrains s.r.o. ("JetBrains") may use my name, email address, and location data to send me newsletters, including commercial communications, and to process my personal data for this purpose. I agree that JetBrains may process said data using third-party service providers.

Linux (.tar.gz)

Windows (.exe)

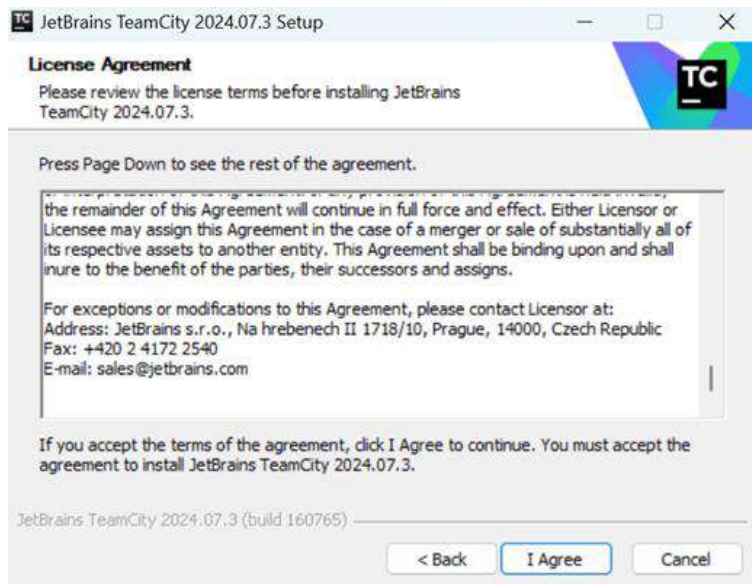
macOS (.tar.gz)

[Download](#) [Windows \(.exe\)](#)

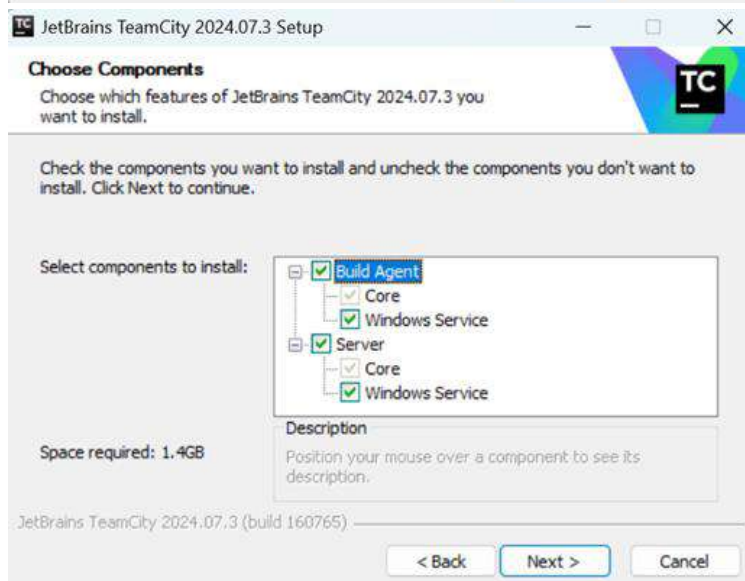
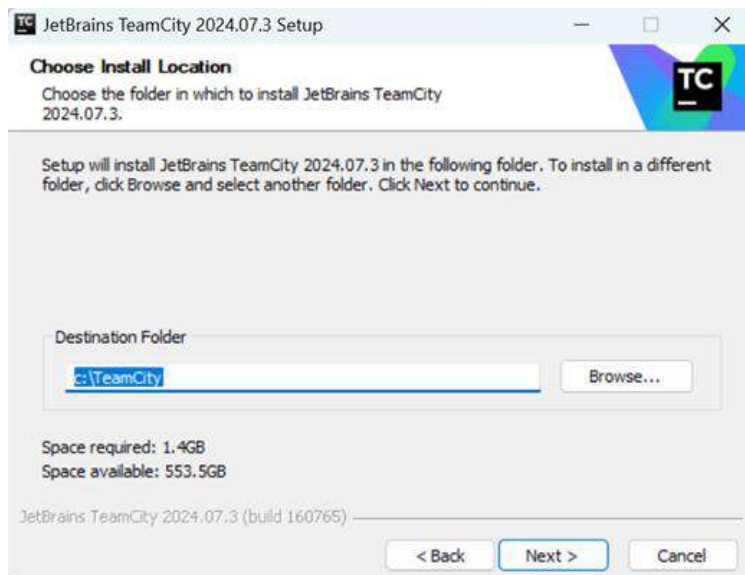
- Run the application to install the TeamCity server in your environment.

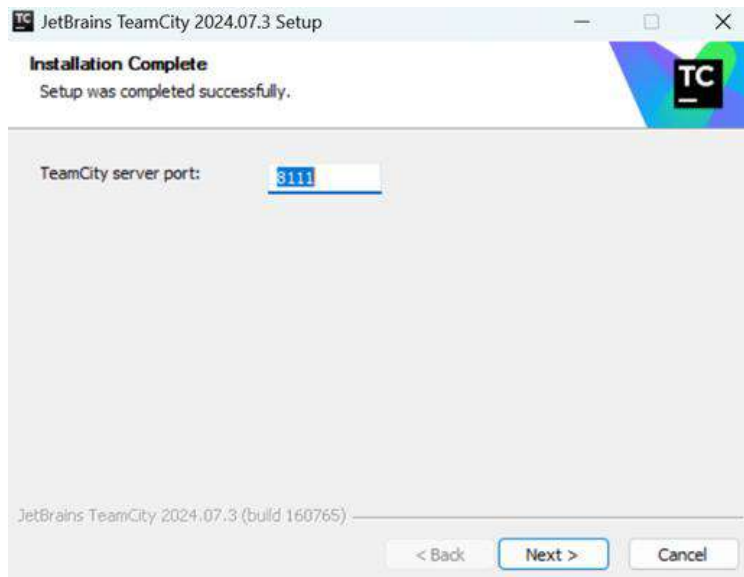


- Accept the terms and conditions and click "I Agree".

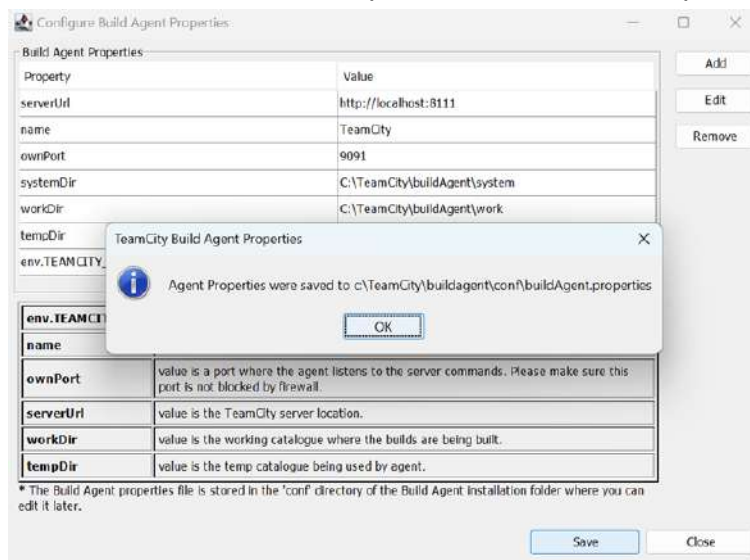


- Keep the default settings for easier integration and management of the server.

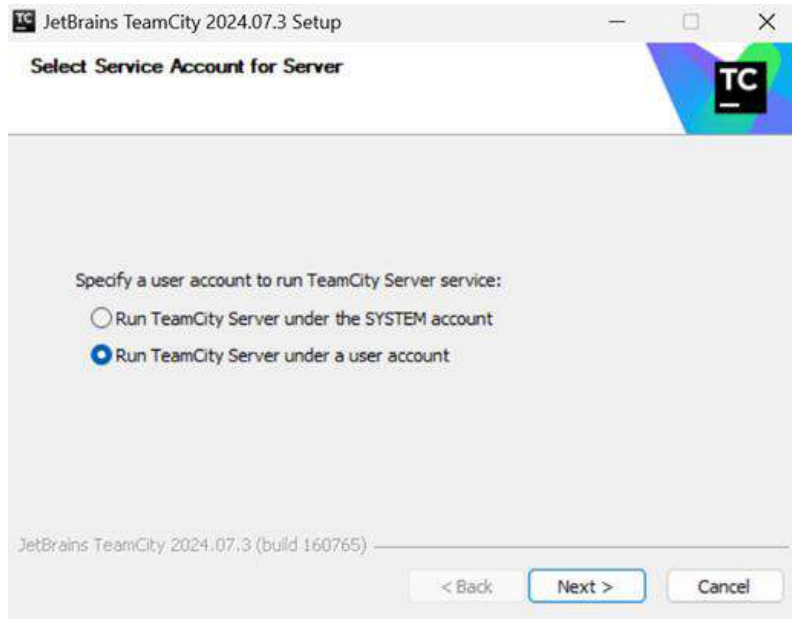




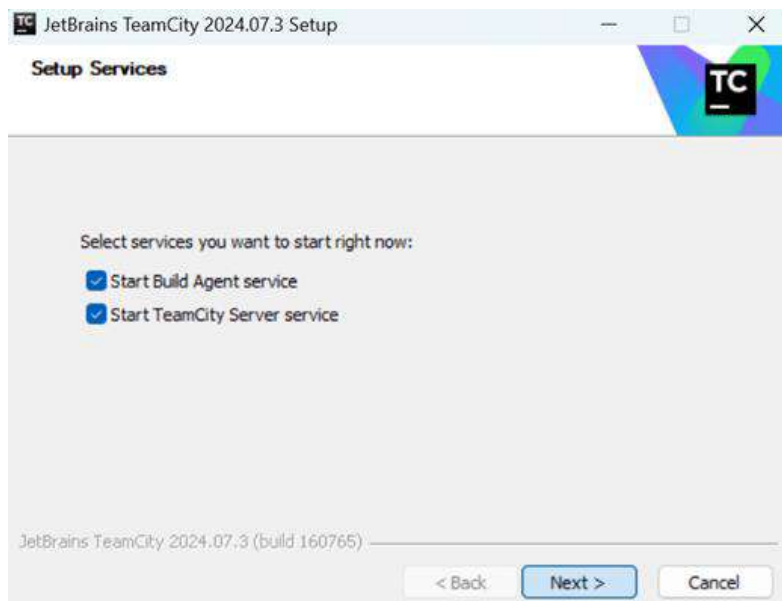
- Click Save and then OK to proceed to the next step of account creation.



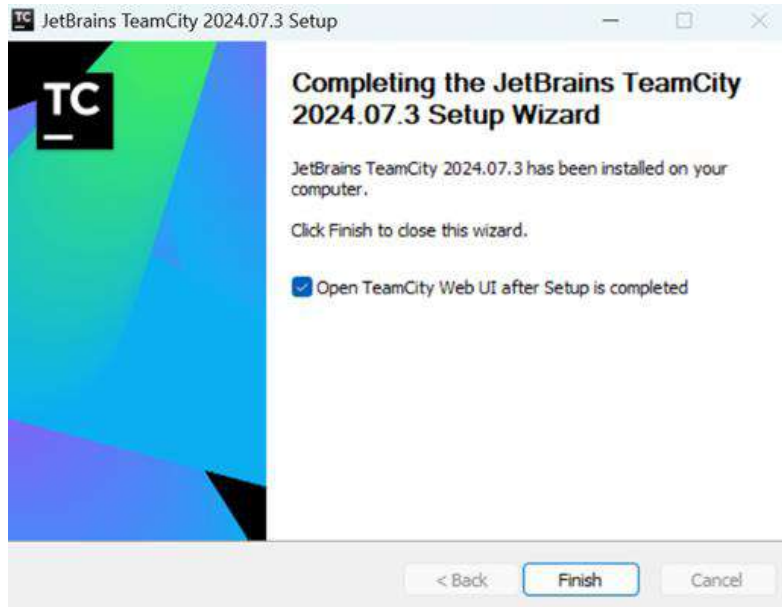
- Select the option to run the TeamCity Server and TeamCity Agent under the user account and click Next. It will ask you to login as the current user of your system.



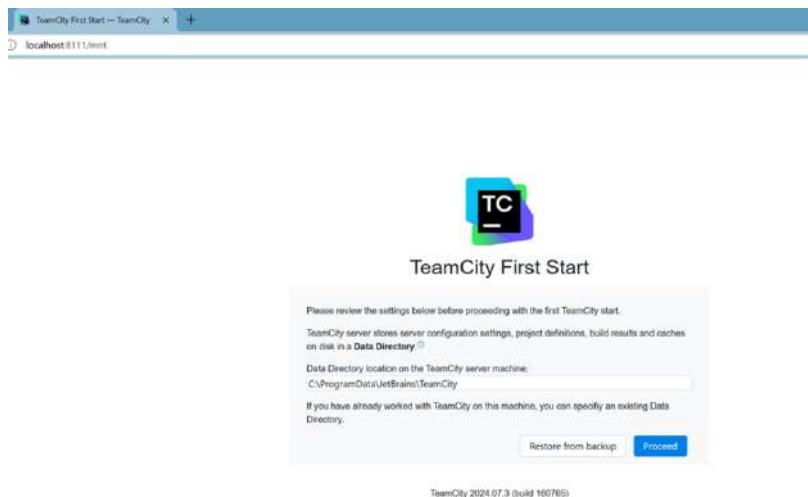
- Select both the options to start the build agent service and the TeamCity server service. Click Next.



- Click Finish. This will take you the Web UI of the TeamCity where you can create your projects and set the configurations policies.



- The Web UI runs on localhost on port 8111 (default). Click Proceed to set the location of data directory for TeamCity server. Access TeamCity from a browser by going to <http://localhost:8111>



- Select the appropriate database which will store the build history and your account related information. We will be choosing HSQLDB database for this project.



The screenshot shows the 'Database connection setup' window in TeamCity. At the top is the TeamCity logo. Below it, the title 'Database connection setup' is centered. The main content area has a light blue background. It starts with the text 'TeamCity server stores builds history and users-related data in an SQL database.' followed by a link icon. Then, there's a section 'Select the database type:' with a dropdown menu. The dropdown is open, showing options: 'Internal (HSQLDB)' (selected), 'PostgreSQL', 'MySQL', 'Oracle', and 'MS SQL Server'. To the left of the dropdown, there's explanatory text: 'The internal database suits evaluation and is not intended for production. We strongly recommend using an external database in a production environment.' and 'You can start with the internal database to migrate the data to an external one after successful evaluation.' At the bottom right of the main area is a blue 'Proceed' button. Below the main area, the version 'TeamCity 2024.07.3 (build 160765)' is displayed.

TeamCity server stores builds history and users-related data in an SQL database.

Select the database type: Internal (HSQLDB) PostgreSQL MySQL Oracle MS SQL Server

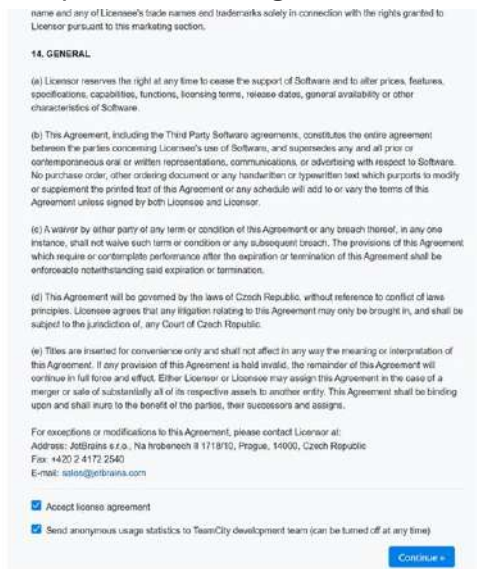
The internal database suits evaluation and is not intended for production. We strongly recommend using an external database in a production environment.

You can start with the internal database to migrate the data to an external one after successful evaluation.

Proceed

TeamCity 2024.07.3 (build 160765)

- Accept the license agreement and click Continue.



The screenshot shows the 'License Agreement' screen. It has a light blue background. At the top, there's a line of small text: 'name and any of Licensee's trade names and trademarks solely in connection with the rights granted to Licensee pursuant to this marketing section.' Below this is the section header '14. GENERAL'. There are five paragraphs of text, each starting with a letter in parentheses: (a) Licensee reserves the right at any time to cease the support of Software and to alter prices, features, specifications, capabilities, functions, licensing terms, release dates, general availability or other characteristics of Software. (b) This Agreement, including the Third Party Software agreements, constitutes the entire agreement between the parties concerning Licensee's use of Software, and supersedes any and all prior or contemporaneous oral or written representations, communications, or advertising with respect to Software. No purchase order, other ordering document or any handwritten or typewritten text which purports to modify or supplement the printed text of this Agreement or any schedule will add to or vary the terms of this Agreement unless signed by both Licensee and Licensor. (c) A waiver by either party of any term or condition of this Agreement or any breach thereof, in any one instance, shall not waive such term or condition or any subsequent breach. The provisions of this Agreement which require or contemplate performance after the expiration or termination of this Agreement shall be enforceable notwithstanding said expiration or termination. (d) This Agreement will be governed by the laws of Czech Republic, without reference to conflict of laws principles. Licensee agrees that any litigation relating to this Agreement may only be brought in, and shall be subject to the jurisdiction of, any Court of Czech Republic. (e) Titles are inserted for convenience only and shall not affect in any way the meaning or interpretation of this Agreement. If any provision of this Agreement is held invalid, the remainder of this Agreement will continue in full force and effect. Either Licensor or Licensee may assign this Agreement in the case of a merger or sale of substantially all of its respective assets to another entity. This Agreement shall be binding upon and shall inure to the benefit of the parties, their successors and assigns. Below the paragraphs is contact information: 'For exceptions or modifications to this Agreement, please contact Licensor at: Address: Jotifraim s.r.o., Na Irabenech 8 171810, Prague, 14000, Czech Republic Fax: +420 2 4172 2540 E-mail: sales@jotifraim.com'. At the bottom, there are two checkboxes: 'Accept license agreement' (checked) and 'Send anonymous usage statistics to TeamCity development team (can be turned off at any time)' (checked). A blue 'Continue' button is at the bottom right.

name and any of Licensee's trade names and trademarks solely in connection with the rights granted to Licensee pursuant to this marketing section.

14. GENERAL

(a) Licensee reserves the right at any time to cease the support of Software and to alter prices, features, specifications, capabilities, functions, licensing terms, release dates, general availability or other characteristics of Software.

(b) This Agreement, including the Third Party Software agreements, constitutes the entire agreement between the parties concerning Licensee's use of Software, and supersedes any and all prior or contemporaneous oral or written representations, communications, or advertising with respect to Software. No purchase order, other ordering document or any handwritten or typewritten text which purports to modify or supplement the printed text of this Agreement or any schedule will add to or vary the terms of this Agreement unless signed by both Licensee and Licensor.

(c) A waiver by either party of any term or condition of this Agreement or any breach thereof, in any one instance, shall not waive such term or condition or any subsequent breach. The provisions of this Agreement which require or contemplate performance after the expiration or termination of this Agreement shall be enforceable notwithstanding said expiration or termination.

(d) This Agreement will be governed by the laws of Czech Republic, without reference to conflict of laws principles. Licensee agrees that any litigation relating to this Agreement may only be brought in, and shall be subject to the jurisdiction of, any Court of Czech Republic.

(e) Titles are inserted for convenience only and shall not affect in any way the meaning or interpretation of this Agreement. If any provision of this Agreement is held invalid, the remainder of this Agreement will continue in full force and effect. Either Licensor or Licensee may assign this Agreement in the case of a merger or sale of substantially all of its respective assets to another entity. This Agreement shall be binding upon and shall inure to the benefit of the parties, their successors and assigns.

For exceptions or modifications to this Agreement, please contact Licensor at:
Address: Jotifraim s.r.o., Na Irabenech 8 171810, Prague, 14000, Czech Republic
Fax: +420 2 4172 2540
E-mail: sales@jotifraim.com

☒ Accept license agreement

☒ Send anonymous usage statistics to TeamCity development team (can be turned off at any time)

Continue

- Now create a login account for your CI/CD pipeline and projects.



Create Administrator Account

Username

Password

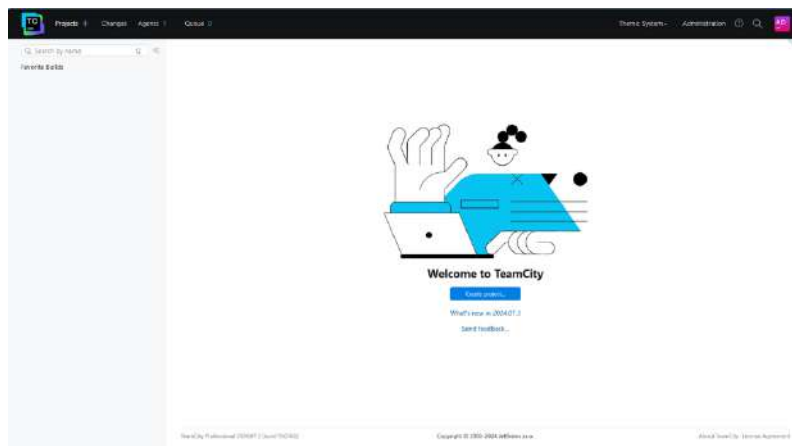
Confirm password

[Login as Super user](#)

Version 2024.07.3 (build 160765)

Step 2: Create a new Project

- After successful login, it will take you to the TeamCity Dashboard. Now you can begin with creating your first project. Click on "Create project".



- There are two choices when you create a project on TeamCity. It can be done either using a repository account or manually.

Let's first see the Manual process. Enter the required details for your project.

- Once successfully created, it will then prompt you to create a build configuration.

Administration / All open projects / New Project

Project "New Project" has been successfully created. You can now create a build configuration.

Name: New Project
Project ID: NewProject
Description: This is a new project.
Default template: The default template selected. The repository will be attached to all build configurations in this project and to other builds or files included in a subproject.

Show advanced options

Create

Build Configurations (00 left)
Build configurations define how to retrieve and build sources of a project.
Create build configuration

Build Configuration Templates
Build configuration templates define settings that can be reused by different build configurations.
Create template

Subprojects
Subprojects can be used to group build configurations and define projects hierarchy within a single project.

- Here as well, you have two options - repository or manual process. We will again proceed with the manual process. Enter the required details and click Create.

Administration / All open projects / New Project / Create Build Configuration

From a repository URL | Manually

Current project: New Project

Name: NewConfiguration
Build configuration ID: NewProject/NewConfiguration
Description: Settings from this project.

Show advanced options

Create

TeamCity Professional 2020.2 build system
Copyright © 2009-2024 JetBrains s.r.o. | License agreement

- It will then ask you to enter the details for Version Control Settings. It is an optional step. We will be skipping it for this project.

Administration / All open projects / New Project / Version Control Settings

New VCS Root

Build configuration successfully created. You can now configure VCS roots.

Type of VCS: From repository URL

Repository URL and Authentication:
Repository URL: A VCS repository URL. Supported formats: http(s), ssh, git, etc. as well as URLs in Maven format.
Username: Provide username if access to the repository requires authentication.
Password / access token: Provide password if access to the repository requires authentication.

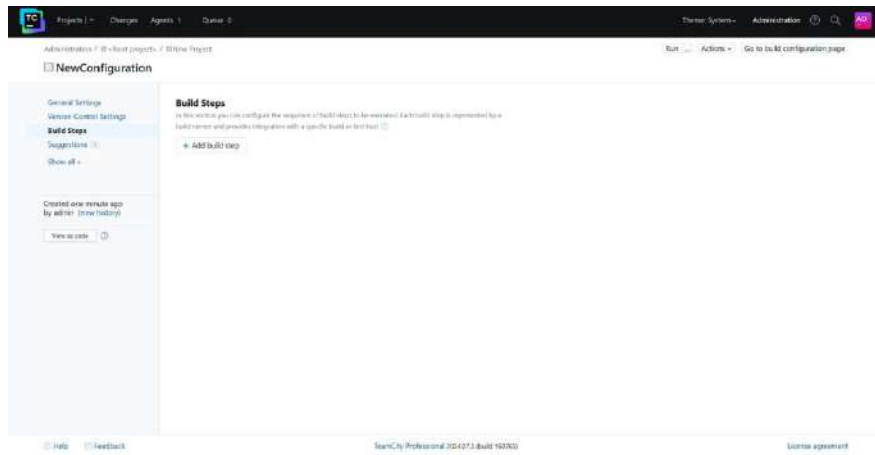
Show advanced options

Create | Skip

TeamCity Professional 2020.2 build system
Copyright © 2009-2024 JetBrains s.r.o. | License agreement

Step 3: Create a Pipeline in your Project

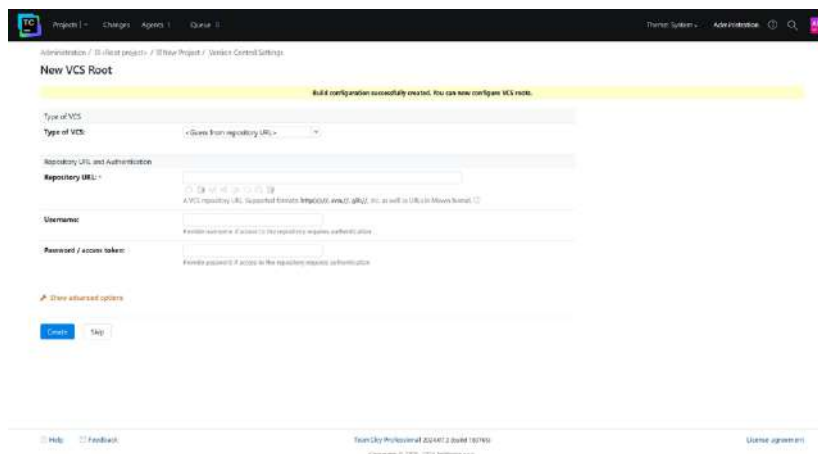
- After setting up the general project and configuration settings, we can now add build steps to our pipeline. These are the processes that will actually execute and perform code signing operations.



- Click on the "Add Build Step" button and choose "Command Line" from the dropdown.



- It will then ask you to enter the details for Version Control Settings. It is an optional step. We will be skipping it for this project.



- Provide the basic details such as Step name and Step ID.

- Select the Custom script code option from the drop-down of the Script field and enter the cmd script to perform the signing operation using the required details.

A sample Command Line script is provided below for your reference:

```
signtool sign /csp "Encryption Consulting Key Storage Provider" /kc <key
name of private key associated with the certificate> /fd <the desired
signing algorithm to be used when signing> /f <The location of the
certificate to be used for signing> /tr <The URL of the time stamping server
to be used> /td <The signing algorithm to be used with the time stamping
server> <The path to the file to be signed>
```

The following are the flags and their meaning in the command:

- /csp: This specifies the Key Service Provider to be used. This should always be "Encryption Consulting Key Storage Provider."

- /kc: This should be the name of the private key associated with your certificate. This will likely be the same name as the certificate name.

- /fd: This is the signing algorithm to be used with the signing function. This can be SHA256, SHA384, or SHA512.

- /f: The name of the certificate to be used for signing. This should be the same name as the /kc flag.

- /tr (optional): This is the URL of the timestamping authority to be used. If /tr is not in use, and timestamping is not required, then /td should also be left out of this command.

- /td (optional): This is the signing algorithm to be used with the timestamping server. This can be SHA256, SHA384, or SHA512. This algorithm should be the same as the /fd field. If /tr is not in use, and timestamping is not required, then /td should also be left out of this command.

- <file path>: This is the name of the file to be signed. This file must be in the directory in which this command is being run.

- The entire step should look something similar to the image below:

New Build Step: Command Line Simple command execution [Change name](#)

Step name:
Optional, specify to distinguish this build step from other steps.

Step ID:
This ID is used in URLs, REST API, DSL, HTTP requests to the server, and configuration settings in the TeamCity Data Directory. Must be unique across all steps of this configuration.

Run:

Custom script: Enter build script content:

```
1 "C:\Program Files (x86)\Windows Kits\10\bin\x86\signtool.exe" sign /csp "Encryption Consulting Key Storage provider" /kc DemoCertificate /
```

A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.

Container Settings

Run step within container:
E.g. ruby2.4. TeamCity will start a container from the specified image and will try to run this build step within this container.

[Show advanced options](#)

- Now repeat the above process to create another build step for Verification of the signed file.

A sample Command Line script is provided below for your reference:

signtool verify <The path to the file which was signed>

- The entire step should look something similar to the image below:

New Build Step: Command Line Simple command execution [Change name](#)

Step name:
Optional, specify to distinguish this build step from other steps.

Step ID:
This ID is used in URLs, REST API, DSL, HTTP requests to the server, and configuration settings in the TeamCity Data Directory. Must be unique across all steps of this configuration.

Run:

Custom script: Enter build script content:

```
1 "C:\Program Files (x86)\Windows Kits\10\bin\x86\signtool.exe" verify /pa "C:\demo files\test files\testfiles.exe"
```

A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.

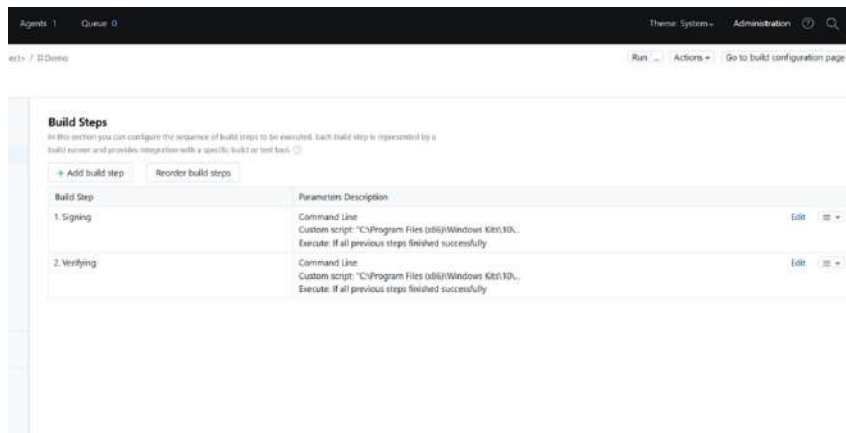
Container Settings

Run step within container:
E.g. ruby2.4. TeamCity will start a container from the specified image and will try to run this build step within this container.

[Show advanced options](#)

Step 4: Run TeamCity Pipeline

- Now, to run the pipeline, click on "Run," which is on the top right of the project view.
- After clicking "Run", you can see the live action of the pipeline by clicking "Agents"(top-left)->Select the agent name->Build History.



- A successful execution of a pipeline will look like this. Here you can see that both the build steps - signing and verification - have been completed.

