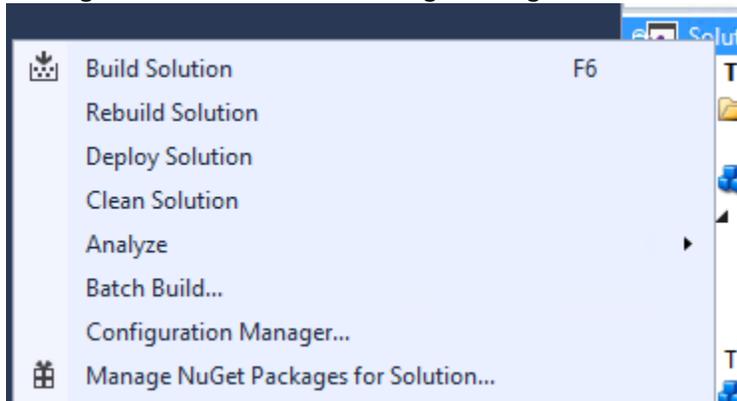


Building in the VSTS cloud with Nuget TDS build package

This is a quick tutorial how to use TDS with Visual Studio online.

1. Create a solution with TDS project(s)
2. Check it in to VSTS
3. There's a prepared NuGet package with the TDS build components that's ready to use - it's located inside the TDS installation zip, under the \NuGetPackages folder
4. Add TDS Nuget build components to the solution
 - a. Right-click on solution -> Manage Packages for Solution



- b. Add custom NuGet repository, pointing at the TDS NuGet package folder -or- in case you have your own NuGet repo, upload the package to it
- c. Switch to the custom repository and select the HedgehogDevelopment.TDS package that holds the build components
- d. Attach the package to **every** TDS project inside the Solution:

Manage Packages for Solution

Browse Installed Updates Consolidate

Package source: Package source

Search (Ctrl+E) Include prerelease

HedgehogDevelopment.TDS by Charles Turano
Adds Team Development for Sitecore (TDS) build components to a TDS project. This removes the need to install TDS on a build server when building TDS projects.

TDS2RocksConnector by Charles Turano v5.5.0.15
Allows TDS and Rocks to communicate

TemplateDiagram by Sean Kearney v1.2.0
Visualize your Sitecore templates in a diagram.

HedgehogDevelopment.TDS

Version(s) - 1

Project	Version
<input checked="" type="checkbox"/> TDS Project	5.5.0.15
<input checked="" type="checkbox"/> TDS Project 2	
<input type="checkbox"/> Web Project	

Installed: 5.5.0.15 Uninstall

Version: Latest stable 5.5.0.15 Install

Options

Description
Adds Team Development for Sitecore (TDS) build components to a TDS project. This removes the need to install TDS on a build server when building TDS projects.

Version: 5.5.0.15
Author(s): Charles Turano
Date published: Monday, June 13, 2016 (6/13/2016)
Tags: TDS

Dependencies
No dependencies

- e. After installing, restart Visual Studio to apply the changes
- f. On VS start, check-in your changes to source control
5. Prepare a build definition on the VSTS build server. Two options here:
 - a. Option 1: Visual Studio Build (don't use "Restore NuGet Packages" because it's deprecated)
 - i. Set a solution target
 - ii. Point at the correct build platform and build configuration
 - iii. Select the VS version you are building against
 - iv. Select the MSBuild Architecture

Definitions / | Builds

Build Options Repository Variables Triggers General Retention History

Save Queue build... Undo

+ Add build step...

- NuGet restore
NuGet Installer
- Build solution
Visual Studio Build

Build solution

Solution: <path to the solution file>

MSBuild Arguments:

Platform: Any CPU

Configuration: Cloud

Clean:

Visual Studio Version: Visual Studio 2015

Advanced

Control Options

Enabled:

Continue on error:

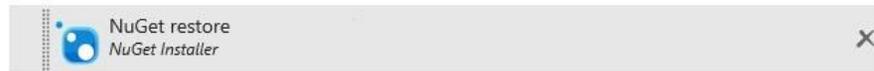
Always run:

Timeout: 0

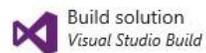
More Information

- v. Add NuGet Installer step
 1. Point it to the correct Solution path
 2. Set Restore as Installation type
 3. Move the Nuget Installer step ahead of the build step

+ Add build step...



NuGet restore
NuGet Installer



Build solution
Visual Studio Build

NuGet restore

Path to solution or packages.config
Path to NuGet.config
Installation type Restore Install
Disable local cache
NuGet arguments

Advanced

Path to NuGet.exe

Control Options

Enabled
Continue on error
Always run
Timeout

[More Information](#)

- b. Option 2: Using an MSBuild step (don't use "Restore NuGet Packages" because it's deprecated)
 - i. Set a solution target
 - ii. Point at the correct build platform and build configuration
 - iii. Select the MSBuild version you want to use
 - iv. Select the MSBuild Architecture

Definitions / Builds

Build Options Repository Variables Triggers General Retention History

Save Queue build... Undo

+ Add build step...

- NuGet restore
NuGet Installer
- Build solution
MSBuild

Build solution

Project: <path to the solution file>

Platform: Any CPU

Configuration: Cloud

MSBuild Arguments:

Clean:

Advanced

Restore NuGet Packages:

Record Project Details:

MSBuild: Version Specify Location

MSBuild Version: MSBuild 14.0

MSBuild Architecture: MSBuild x86

Control Options

Enabled:

Continue on error:

Always run:

Timeout: 0

[More Information](#)

- v. Add NuGet Installer step (same instruction as with Option 1)
- vi. Move the Nuget Installer step ahead of the build step
- c. Prepare the TDS license
 - i. Install the TDS license information as Variables for the build definition - **do NOT use the keychains to encrypt the variables, because the license will not work** - reference to VSTS docs <https://www.visualstudio.com/docs/build/define/variables>
 1. TDS_Owner for company
 2. TDS_Key for the license key

Definitions / Builds

Build Options Repository **Variables** Triggers General Retention History

Save Queue build... Undo

List of predefined variables

Name	Value	Allow at Queue Time
system.collectionId		<input type="checkbox"/>
system.teamProject		<input type="checkbox"/>
system.definitionId		<input type="checkbox"/>
X TDS_Owner		<input type="checkbox"/>
X TDS_Key		<input type="checkbox"/>
+ Add variable		<input type="checkbox"/>

If the license is not accepted, you'll get the following exception: Exception Invalid License(Exception) in file sitecore\content.item.