

# ComponentSpace

## SAML for ASP.NET Core

### Installation Guide

## Contents

Introduction .....	1
Installation .....	1
NuGet Package Source .....	1
NuGet.config .....	1
NuGet Package Manager .....	2
NuGet Package Installation .....	3
Evaluation vs Licensed Version .....	5
Upgrading to a Licensed Version .....	6
Uninstallation .....	7

## Introduction

The ComponentSpace SAML for ASP.NET Core is a .NET standard class library that provides SAML v2.0 assertions, protocol messages, bindings and profiles functionality.

You can use this functionality to easily enable your ASP.NET Core web applications to participate in SAML v2.0 federated single sign-on (SSO) either as an Identity Provider (IdP) or Service Provider (SP). Example applications with full source code are included.

## Installation

The product is supplied as a ZIP file. This includes the binaries, documentation and example projects. Simply unzip to a convenient location on your file system (eg C:\SAML for .NET Core).

The product does not need to be installed on production or test servers. Instead, the SAML for ASP.NET Core class library should be deployed along with the application.

## NuGet Package Source

The ComponentSpace.Saml2 DLL is shipped as a NuGet package. This is found in the bin folder and its name format is:

ComponentSpace.Saml2.<version-number>.nupkg

Note that an evaluation version of the product is published through nuget.org.

During evaluation, the nuget.org source may be used. Alternatively, a local package source may be specified in Visual Studio.

For general information on NuGet packages and their consumption, refer to:

<https://docs.microsoft.com/en-us/nuget/consume-packages/overview-and-workflow>

For information on hosting packages, refer to:

<https://docs.microsoft.com/en-us/nuget/hosting-packages/overview>

For information on using Azure Artifacts in an Azure DevOps environment, refer to:

<https://docs.microsoft.com/en-us/azure/devops/artifacts/get-started-nuget>

## NuGet.config

A NuGet.config file is included with the examples solution. It specifies the bin folder as a package source.

For new solutions, a similar NuGet.config file may be used to specify the package source for that solution.

NuGet includes a number of options for specifying the package source including project-specific, solution-specific and machine-wide NuGet.config files. The appropriate file should be updated to specify the package source.

For more information, refer to:

<https://docs.microsoft.com/en-us/nuget/consume-packages/configuring-nuget-behavior>

## NuGet Package Manager

Rather than directly updating NuGet.config, the package source may be specified through Visual Studio.

To specify a package source:

1. From Visual Studio's menu, select Tools > Options.
2. Under the listed options, select NuGet Package Manager > Package Sources.

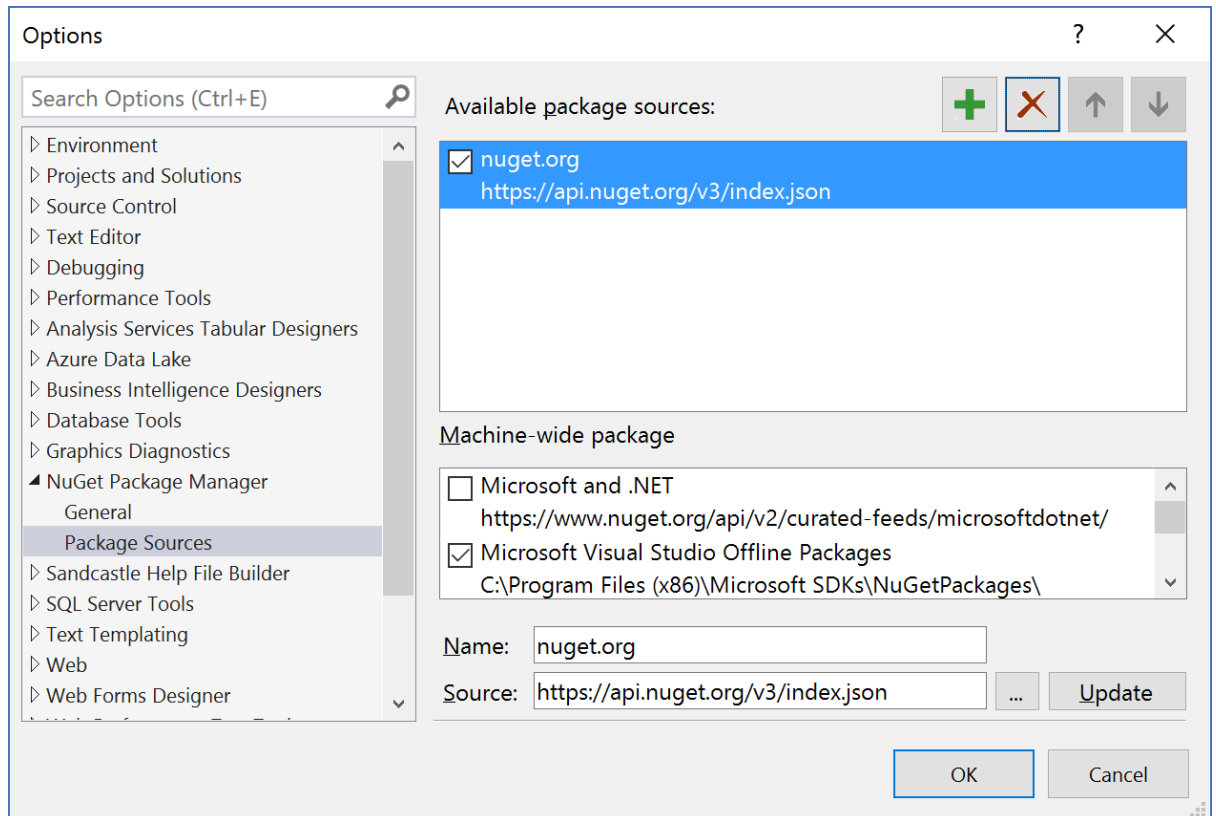


Figure 1 Package Sources

3. Add a package source specifying the bin folder where the ComponentSpace.Saml2 nupkg is located. This package source is available to all ASP.NET Core web application projects.

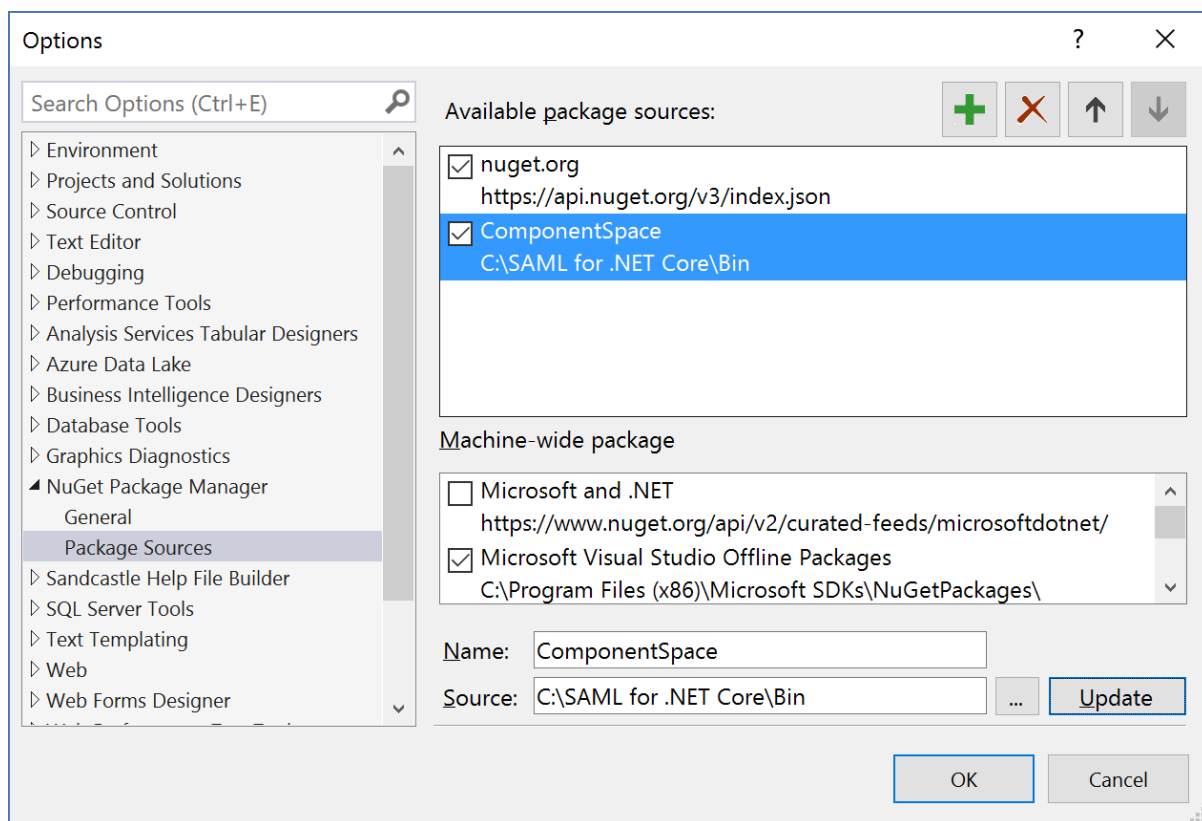


Figure 2 Add ComponentSpace Package Source

## NuGet Package Installation

1. Within an ASP.NET Core web application project, click Manage NuGet Packages and select ComponentSpace as the package source.

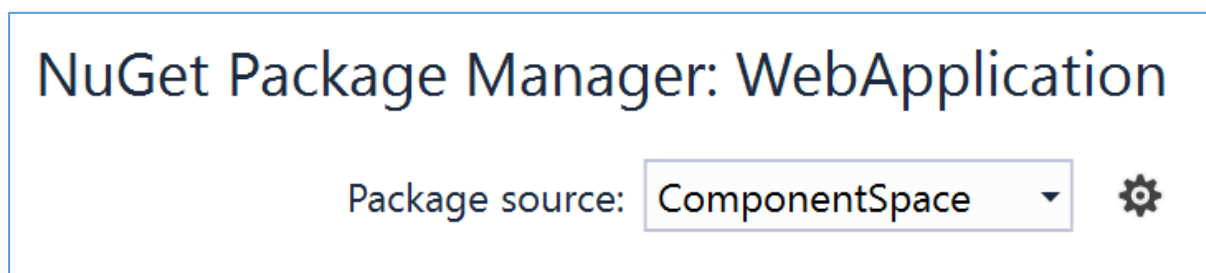


Figure 3 Select ComponentSpace Package Source

2. Browse and install ComponentSpace.Saml2 into the application.

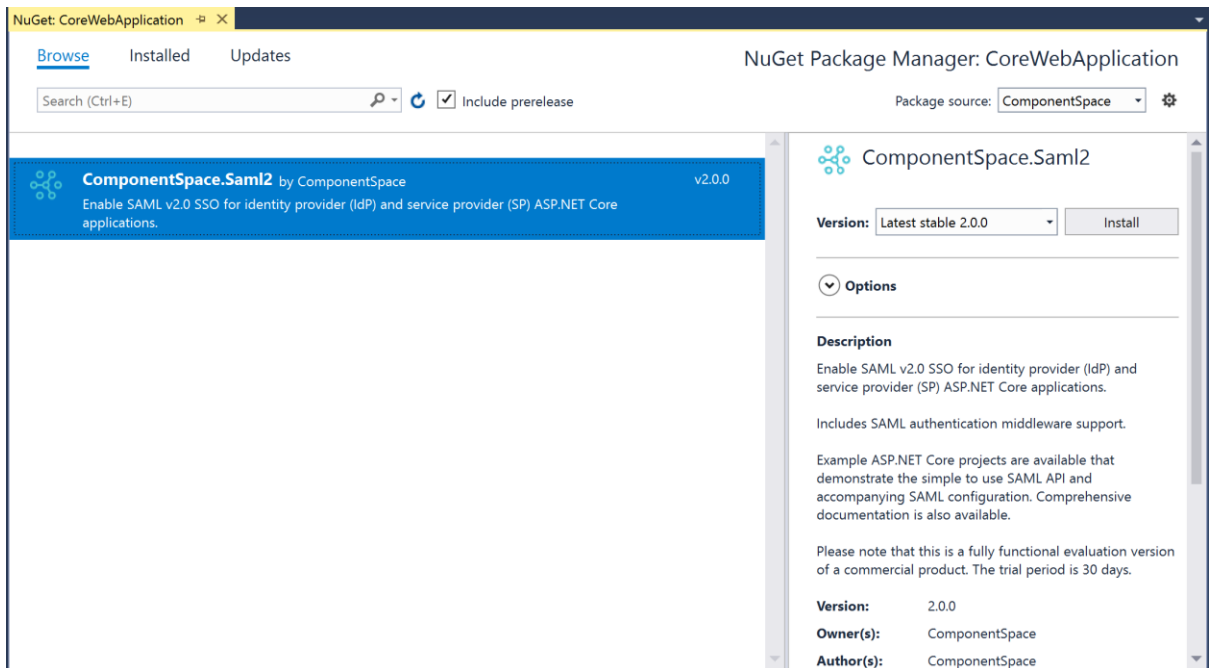


Figure 4 Install ComponentSpace.Saml2

3. Accept the license agreement.

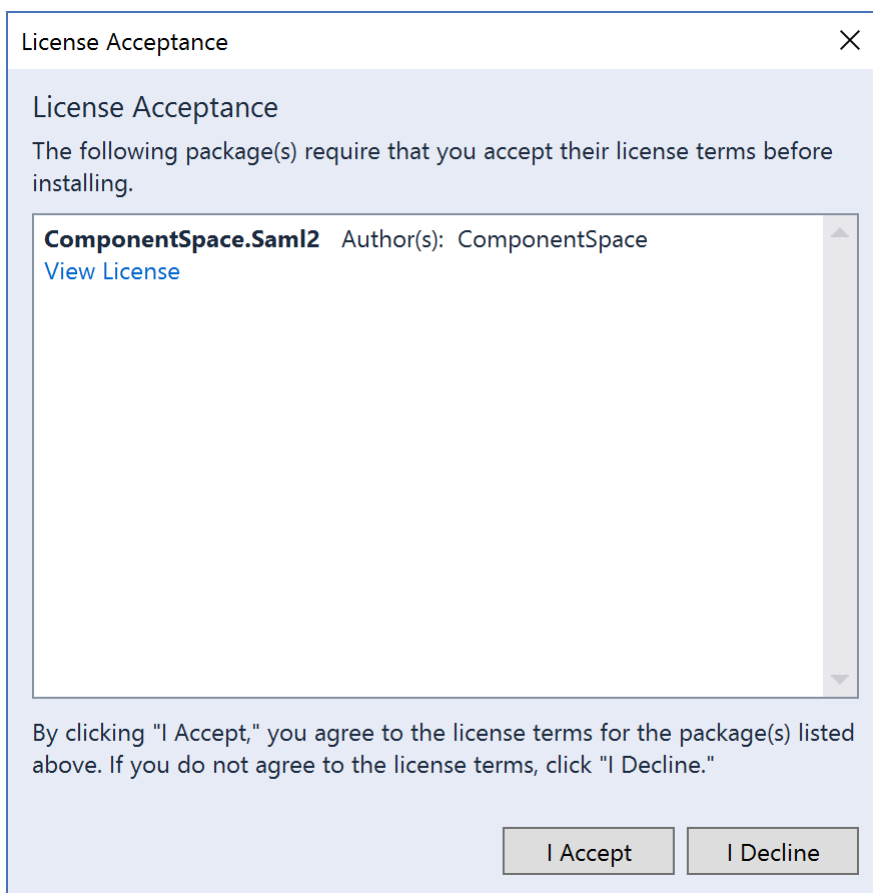


Figure 5 License Agreement Acceptance

- Visual Studio will install the package and its dependencies. This can be confirmed by ensuring Intellisense recognizes the namespace `ComponentSpace.Saml2`.

## Evaluation vs Licensed Version

An evaluation version of the product is equivalent to the licensed version except that it will stop working once the evaluation period expires.

The package name of the evaluation version is **ComponentSpace.Saml2**.

The package name of the licensed version is **ComponentSpace.Saml2.Licensed**.

Different package names are used to avoid accidentally upgrading from a licensed to an evaluation version of the product.

An evaluation version of the ComponentSpace.Saml2 DLL includes the word evaluation. The licensed version doesn't.

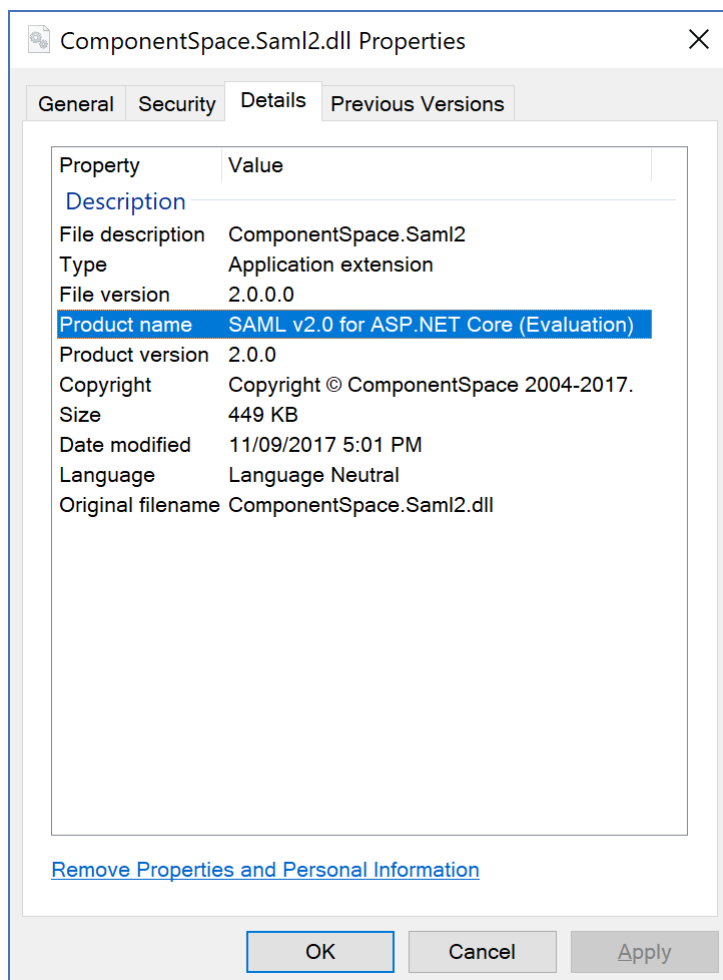




Figure 6 Evaluation Version DLL Properties

Also, the package description specifies whether it's an evaluation or licensed version.

 ComponentSpace.Saml2

**Version:** Latest stable 2.0.0 Install

---

 **Options**

---

**Description**

Enable SAML v2.0 SSO for identity provider (IdP) and service provider (SP) ASP.NET Core applications.

Includes SAML authentication middleware support.

Example ASP.NET Core projects are available that demonstrate the simple to use SAML API and accompanying SAML configuration. Comprehensive documentation is also available.

Please note that this is a fully functional evaluation version of a commercial product. The trial period is 30 days.

**Version:** 2.0.0

**Owner(s):** ComponentSpace

**Author(s):** ComponentSpace

Figure 7 Evaluation Version Package Description

## Upgrading to a Licensed Version

After purchasing a licensed version of the product, ensure all time-limited evaluation packages (i.e. ComponentSpace.Saml2) are replaced with the licensed package (i.e. ComponentSpace.Saml2.Licensed).

Uninstall the **ComponentSpace.Saml2** package from your applications.



## ComponentSpace SAML for ASP.NET Core Installation Guide

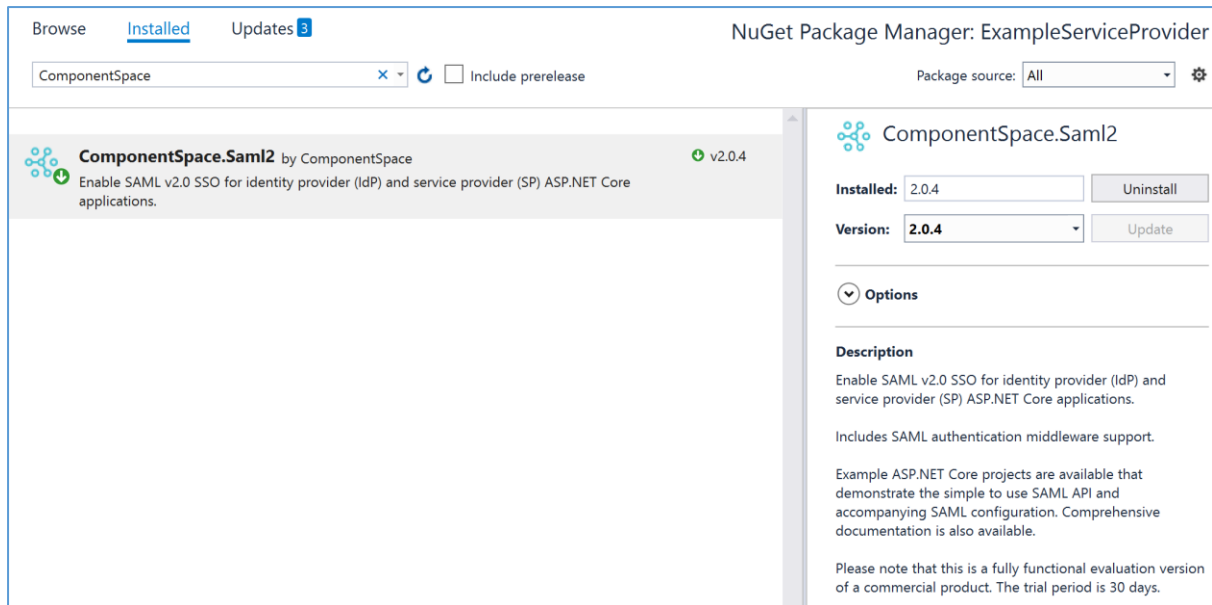


Figure 8 Evaluation Version Installed

Setup a NuGet package source for the licensed package.

Install the **ComponentSpace.Saml2.Licensed** package into your applications.

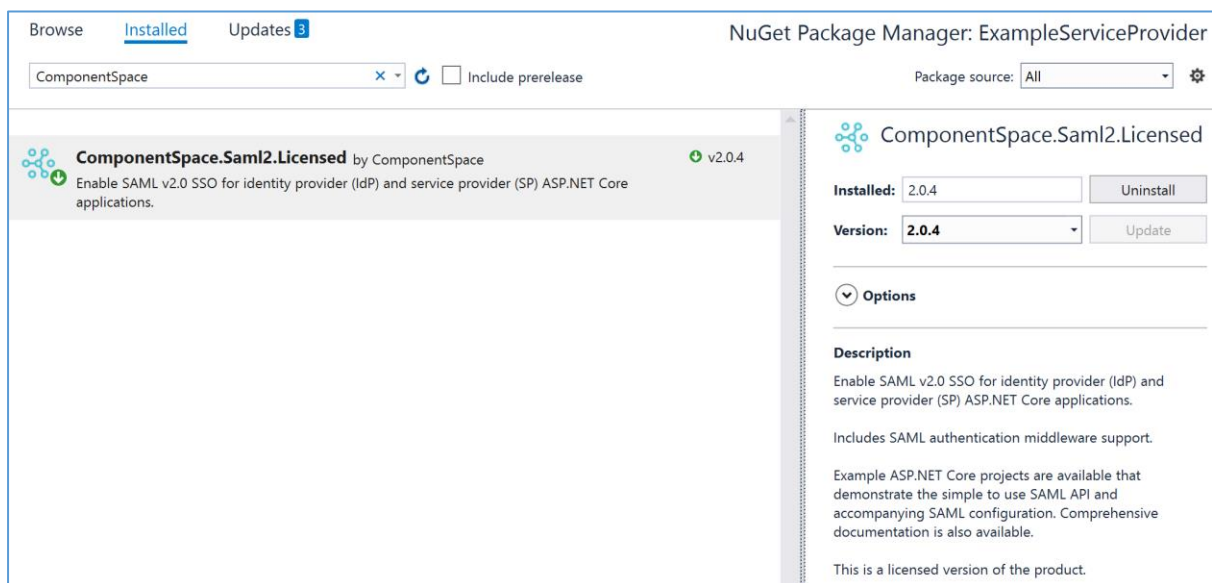


Figure 9 Licensed Version Installed

Your applications should build successfully without having to change the source code.

Be careful not to replace a licensed version of the product with an evaluation version from nuget.org.

## Uninstallation

1. Uninstall the ComponentSpace.Saml2 package from any ASP.NET Core web application projects.
2. Remove the ComponentSpace NuGet package source.
3. Delete the unzipped contents from the file system.