UNISYS | Securing Your Tomorrow®

Agile Business Suite

How to Set up Azure DevOps Services with AB Suite

Applies to: Release 8.0

January 2022

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THIS DOCUMENT. Any product or related information described herein is only furnished pursuant and subject to the terms and conditions of a duly executed agreement to purchase or lease equipment or to license software. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, special, or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Notice to U.S. Government End Users: This software and any accompanying documentation are commercial items which have been developed entirely at private expense. They are delivered and licensed as commercial computer software and commercial computer software documentation within the meaning of the applicable acquisition regulations. Use, reproduction, or disclosure by the Government is subject to the terms of Unisys' standard commercial license for the products, and where applicable, the restricted/limited rights provisions of the contract data rights clauses.

Unisys and other Unisys product and service names mentioned herein, as well as their respective logos, are trademarks or registered trademarks of Unisys Corporation. All other trademarks referenced herein are the property of their respective owners.

How to Set up Azure DevOps Services with AB Suite

Introduction

Agile Business Suite (AB Suite ®) supports Azure DevOps Services for source control, build and deploy application. Azure DevOps Services provide an integrated set of services and tools to manage your software projects. Azure Repos provides Team Foundation Version Control (TFVC) for source control of AB Suite model code and Azure Pipelines provides build and release services to support continuous integration and delivery of your applications, developers can configure it to automatically build and deploy AB Suite applications.

This document outlines the recommended steps to configure the Azure DevOps Services for AB Suite, create an individual model database for each developer and guide you through the required settings to allow Visual Studio build definitions to work with AB Suite projects. The steps are almost similar as using Team Foundation Server, except the way to create Team Project and create build definitions in Azure DevOps Services.

Azure DevOps Services is a set of services provided in cloud. It will host your model files in cloud. You should ensure the user account privileges are set correctly to avoid unauthorized access to the repository. This document lists the necessary permissions for configuring Team Project and installing self-hosted build agent. You can get more permission knowledge from set permissions at the project or collection level from Appendix A.

Developers can use their local machine or Azure virtual machine (Azure VM) to access Azure DevOps Services. AB Suite supports both of them and the following steps are applicable to both. If you want to establish your environment in Azure, you should also consider the cost of the resources in Azure. To get knowledge on using Azure DevOps Services, using Azure VM and estimating its cost, prefer to Appendix A.

Note: *Deployment and Runtime Transfer operations from one Azure VM to another Azure VM hosting AB Suite Runtime for Windows are not currently supported. Generated MSI files for the system and reports should instead be copied over to the AB Suite Runtime environment before being deployed using the Runtime Transfer feature running on the Runtime environment VM.*

This document is applicable for AB Suite 7.0 onwards. AB Suite 7.0 Developer uses Visual Studio Professional 2017 version 15.9.7 from MSDN Subscription. The screenshots in this document use AB Suite 7.0.1000.

In this document, "Test-VSTS-01" is a developer machine,"Test-VSTS-02" is a self-hosted build agent and also used for Windows runtime deployment machine, "ABSuitePro" is the Team Project name and "ABSuiteSample" is the AB Suite solution name. The two machine are hosted in Microsoft Azure in East US region. The size of each machine is 8 GB RAM, 127 GB Disk and 15 GB temporary storage and the cost

of the machine is 0.188$ per hour. The Azure VM charge way is pay-as-you-go. You can change it to suit your own deployments.

Configuring Azure DevOps Services for AB Suite

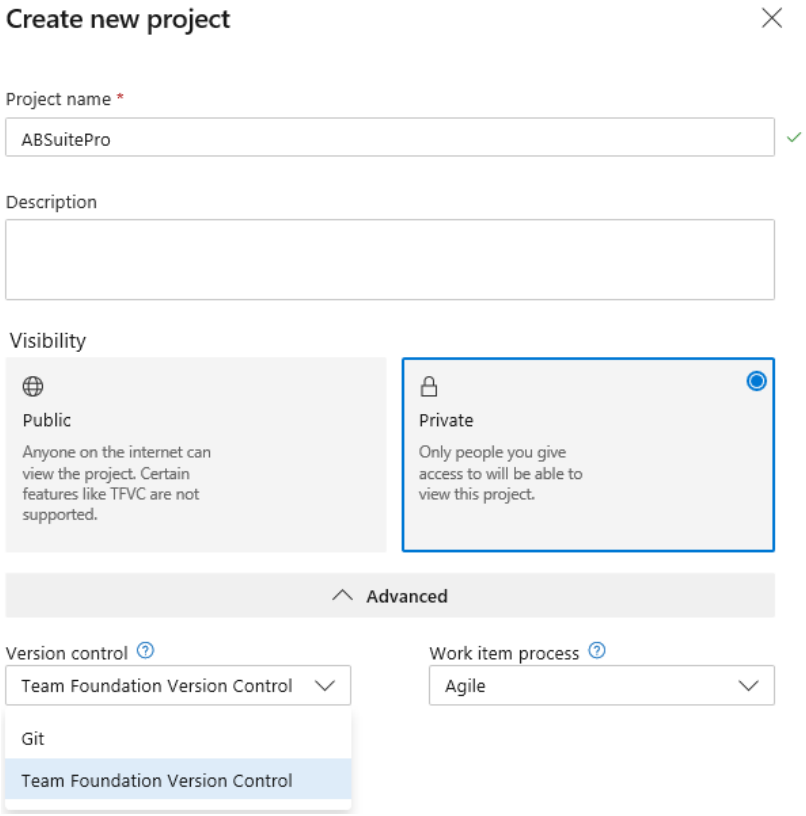
Create a Team Project in Azure DevOps Services

1. Open the Azure DevOps Services website and log in with your account credentials.

   If you do not have an account, you must follow the instructions on the web page to create an account.

2. Click **Create Project** at the right top of page.

   The following window appears.



3. Fill in the required **Project name**, and then select **Private** under Visibility.

   For AB Suite, **Private** is the only option that can be used as **Visibility**, as **Public** does not support TFVC.

4. Expand **Advanced**, and then set **Version control** to **Team Foundation Version Control**.

5. Click **Create**.

   A Team Project named ABSuitePro is created.

Configuring Source Control Settings for AB Suite in Visual Studio

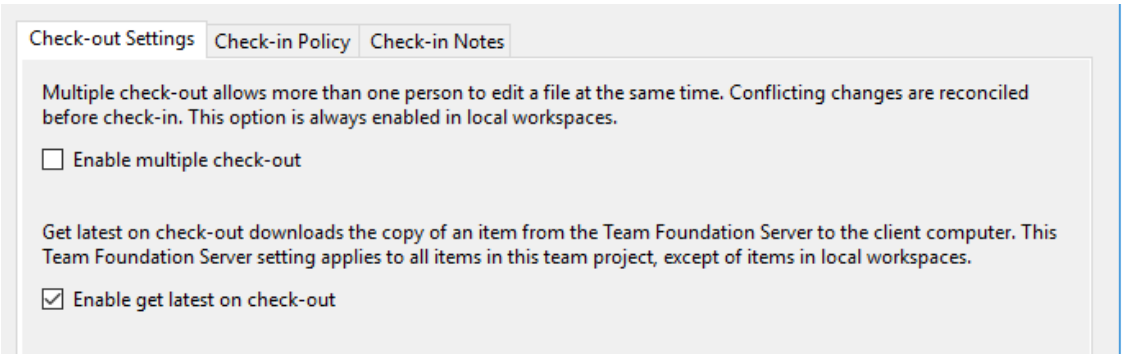
Note: *The Team Project and Team Project Collections configuration changes and settings will affect every developer. It is an operation that only needs to be performed once by a user with AdminProjectRights and AdminConfiguration permissions. For example, an Azure DevOps Services organization owner. You also can refer to Appendix A for accurate permissions in Azure DevOps Services.*

Open Visual Studio, connect to Azure DevOps Services Team Project and then perform the following:

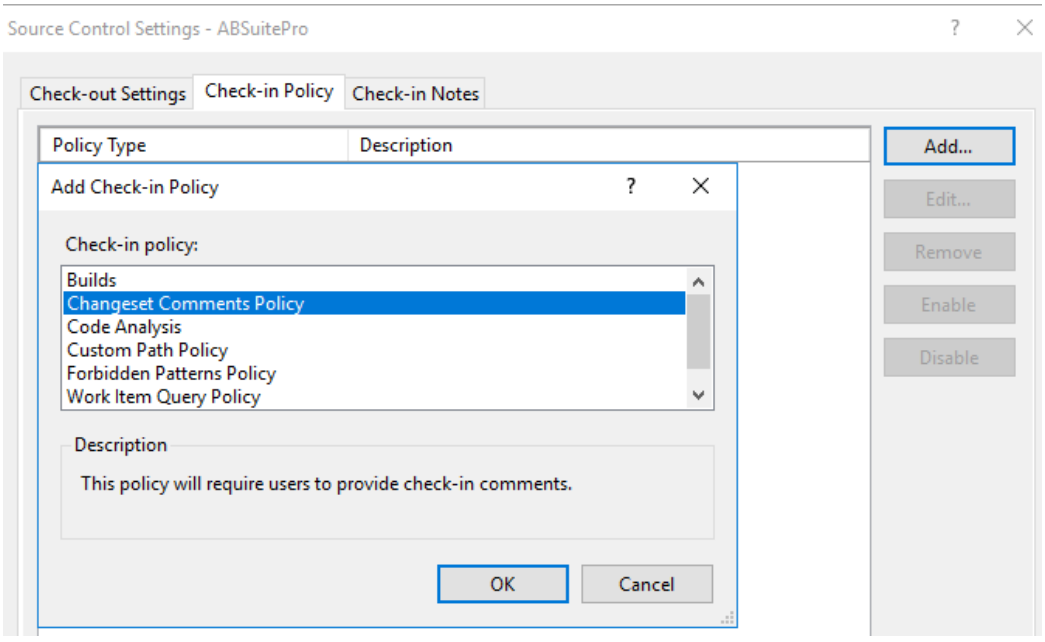
1. Navigate to **Team** > **Team Project Settings** > **Source Control**.

   The **Source Control Settings** dialog box for Team Project appears.

2. On the **Check-out Settings** tab, uncheck the **Enable multiple check-out** check box and select the **Enable get latest on check-out** check box.



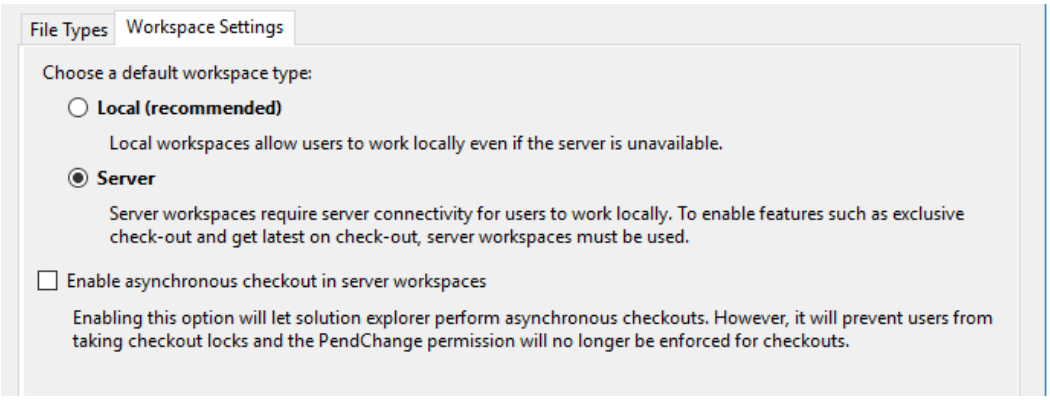
3. On the **Check-in Policy** tab, add the **Changeset Comments Policy** and then click **OK**.



4. Navigate to **Team** > **Team Project Collection Settings** > **Source Control**.

   The **Source Control Settings** dialog box for the Team Project collection appears.

5. On the **Workspace Settings** tab, select **Server** as the default workspace.



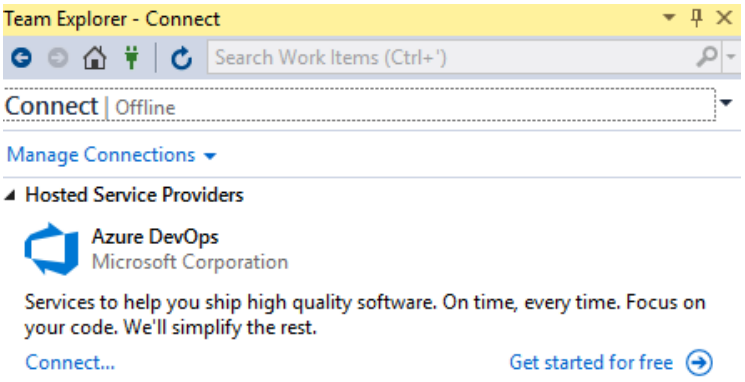
Adding an AB Suite Model to Azure DevOps Services

A user with AdminProjectRights and AdminConfiguration global permissions (for example, an Azure DevOps Services organization owner) needs to add the AB Suite model to Azure DevOps Services before every developer can create their individual model database. To add the model to Azure DevOps Services, perform the following:

Add Project to Azure DevOps Services

1. Open Visual Studio 2017, and then open Team Explorer window.

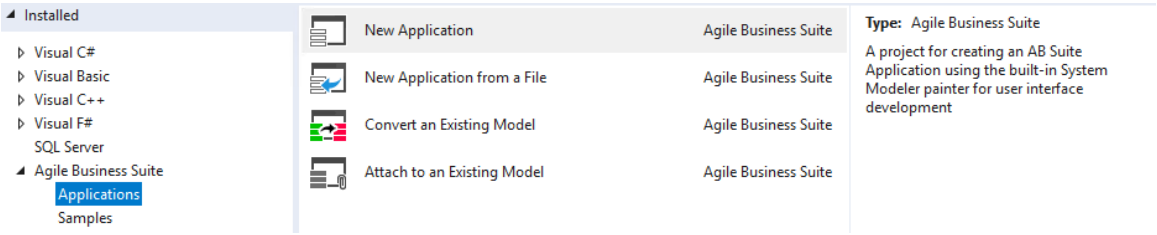
   The following window appears.



2. Click **Connect...**. Click **Add an account**.

   The account should be able to access Azure DevOps Services Team Project.

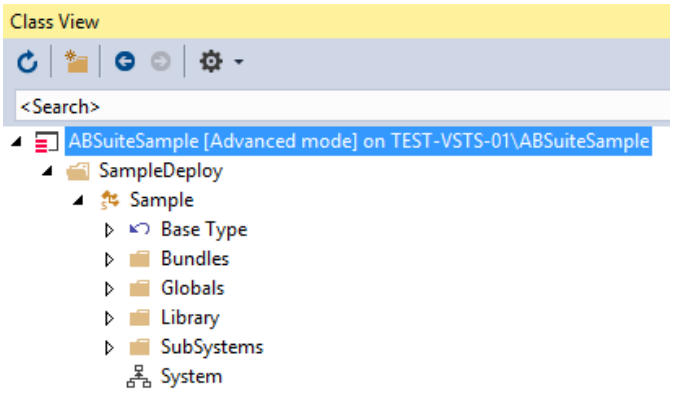
3. You can create a project and model database using one of the new AB Suite 7.0wizards.



4. In this document, it is assumed that you have an existing model database, so you will be using the **Attach to an Existing Model** wizard as an example. Select the existing model, and then click **Next**.

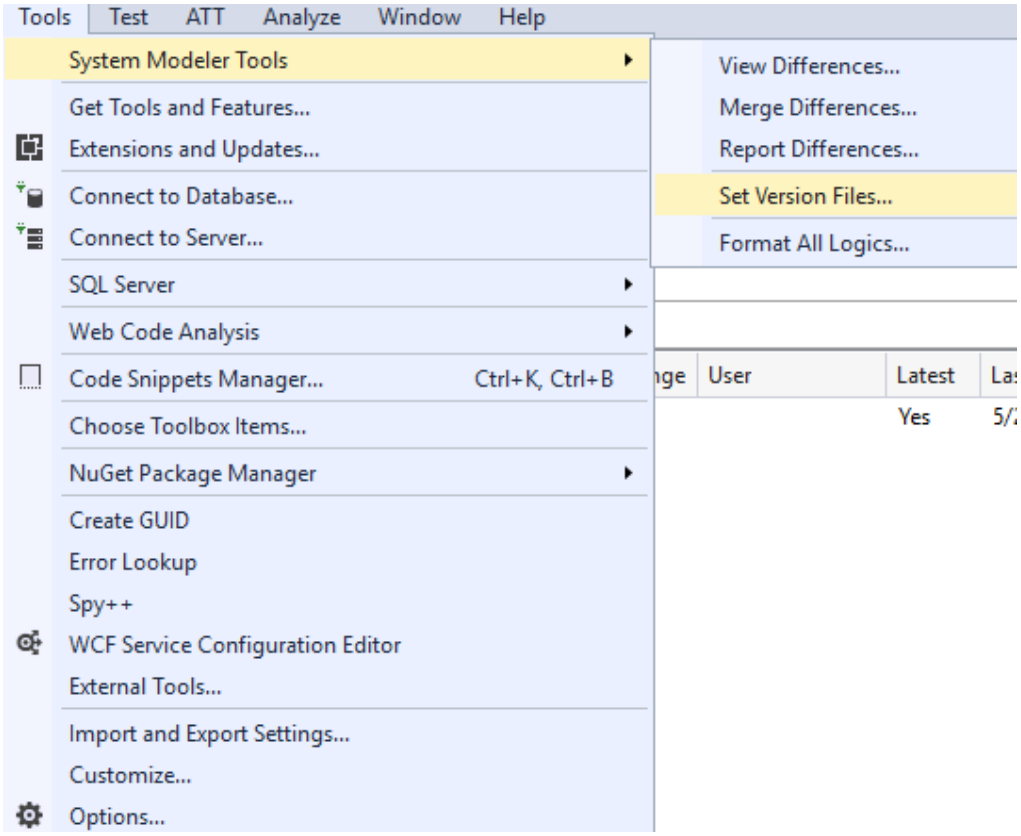
   The following dialog box displays the information of the created project .

5. Click **Finish**.
6. Click **Close** to close the wizard.
7. Open **Class View**, and then select the Model node.

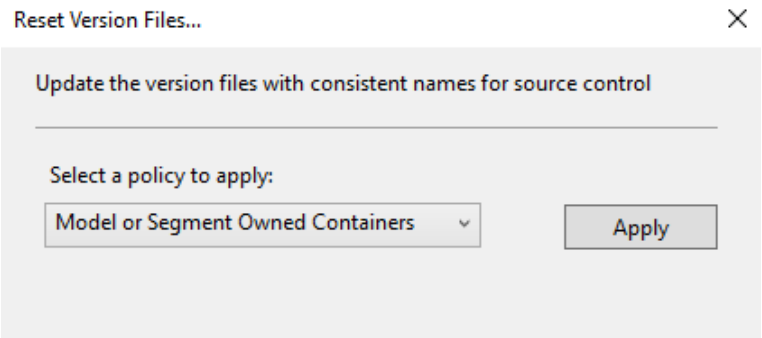


8. Navigate to **Tools** > **System Modeler Tools** > **Set Version Files...**.

   The **Reset Version Files...** dialog box appears.



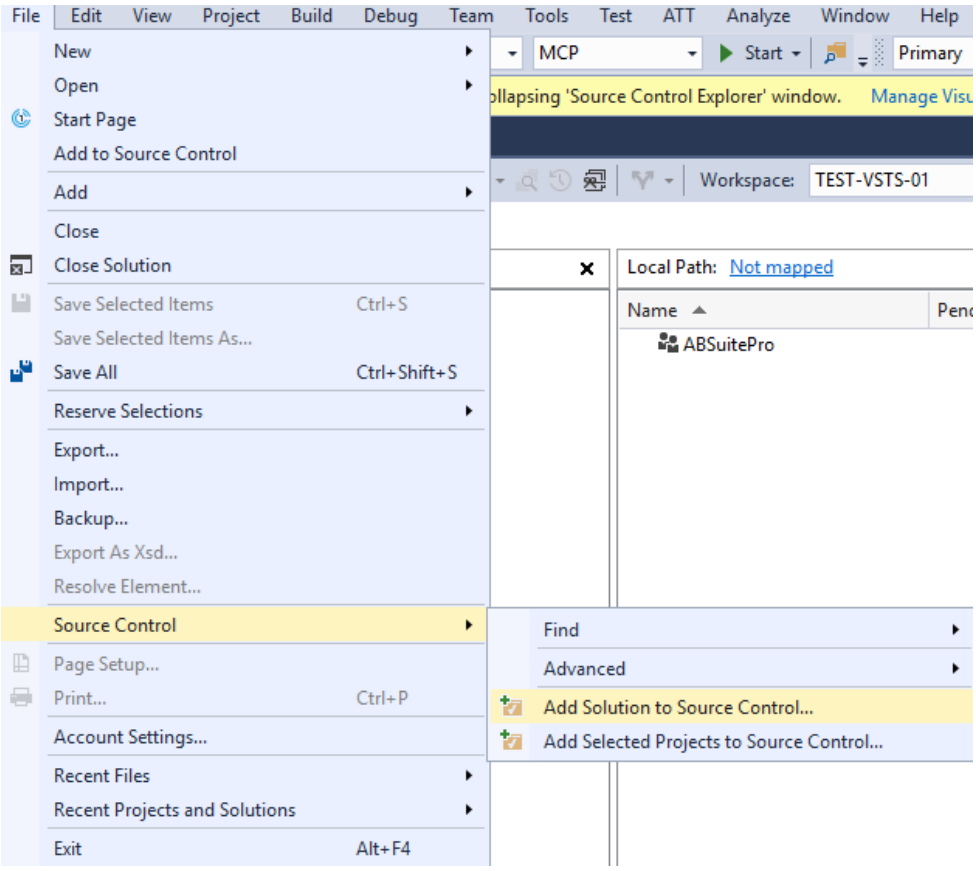
9. You can select the **Model or Segment Owned Containers** policy to generate version files for all classes, methods, dictionaries, diagrams owned by Model or Segment. Click **Apply**.



10. Click **Finish**.

    The project will be reloaded.

11. Navigate to **File** > **Source Control** > **Add Solution to Source Control...**.



12. Select a Team Project (for example, ABSuitePro), type a name for the solution folder and then click **OK**.

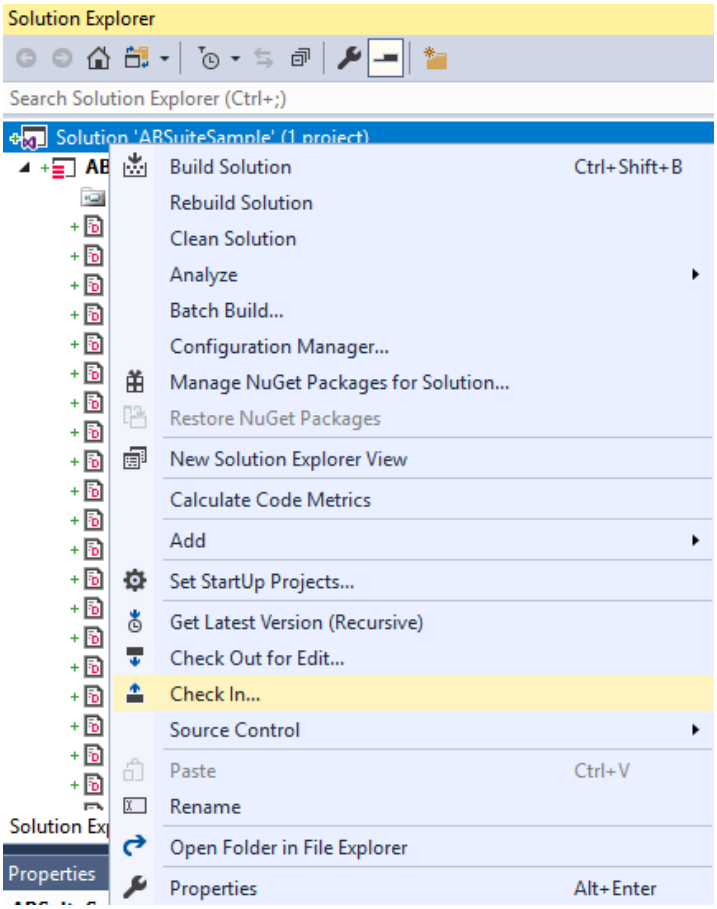
    Note: *You can optionally create a new folder under the selected Team Project by using the* ***Make New Folder*** *button.*

Check In the Pending Changes

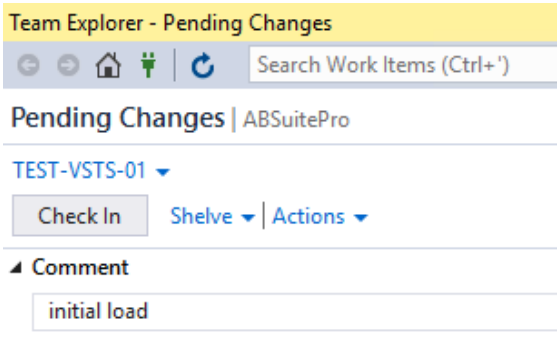
To complete the process, ensure to check-in the pending changes into Azure DevOps Services.

1. Open the **Solution Explorer**, and then select the solution node. Right-click on the solution node, and then click **Check In...**.

   The **Pending Changes** window appears.



2. Enter an appropriate comment in the **Comment** box, and then click **Check In**.



The model is now successfully added to Azure DevOps Services.

Configuring Visual Studio for AB Suite Development

Each developer must configure the user tools to compare and merge changes and also set the appropriate System Modeler policies. Refer to How to configure Visual Studio for AB Suite Development for more details.

Accessing an AB Suite Model Added to Azure DevOps Services

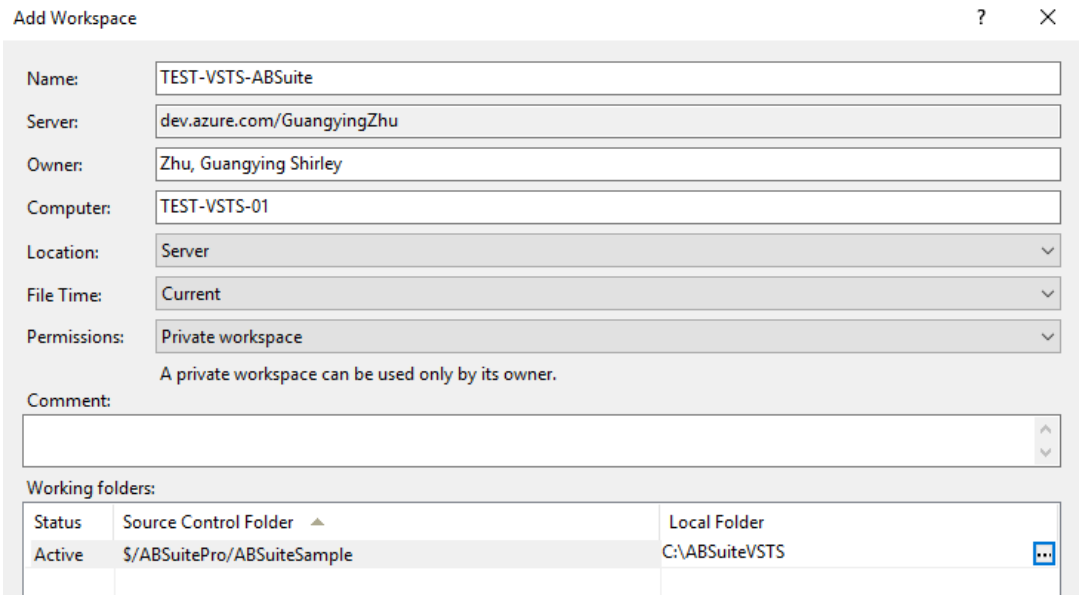
After the model is successfully added to Azure DevOps Services, there are additional steps that must be taken for each developer to setup their own model database. Each developer must either use the "default" workspace or create a workspace to be associated with the Team Project tied to the model database, and mapped to a folder on their machine. The following example makes use of the "default" workspace.

To create a workspace, perform the following:

1. Navigate to the **Source Control Explorer**, and select **Workspaces...** from the Workspace drop-down list.

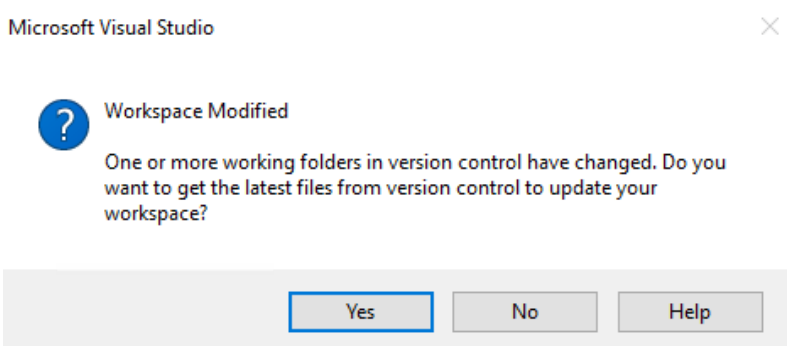
   The **Manage Workspaces** dialog box appears.

2. Select or create a workspace, map it to the correct Team Project folder, and to the desired location on your machine. Ensure your workspace **Location** is set to **Server**.



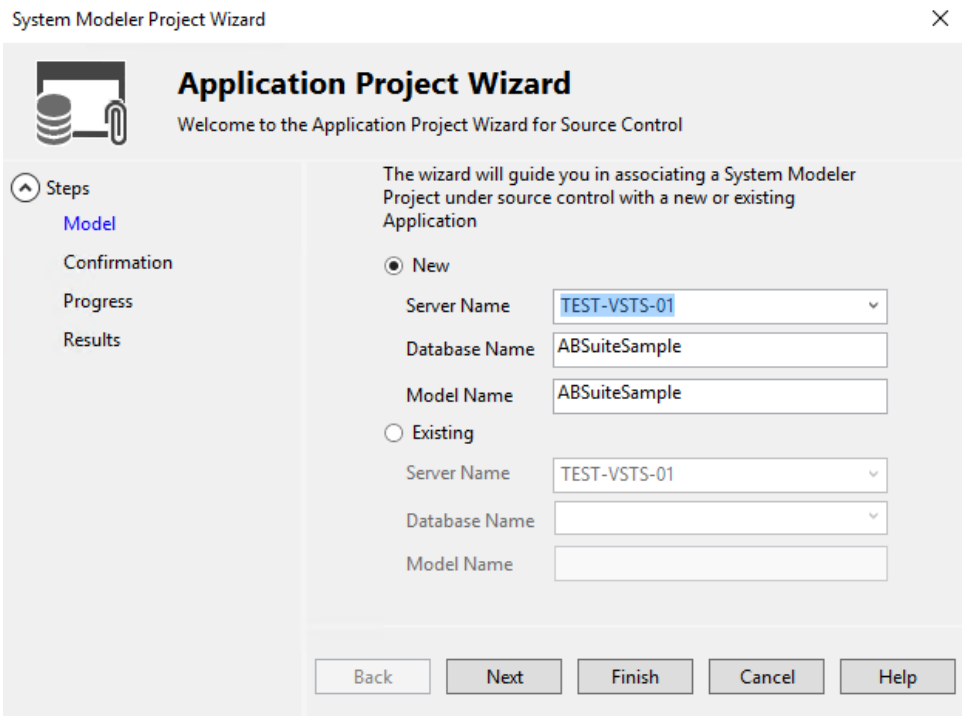
3. Click **OK** to create the workspace.

   The following dialog box appears.



4. Click **Yes** and your workspace will be populated with all of the version files.
5. In the **Source Control Explorer**, again confirm that the correct workspace is selected, and then locate and double-click the ".sln" file to open the solution.

   The **System Modeler Project** wizard appears.
6. Choose **New**, select/enter the instance of SQL Server where you want the model database to be created, enter the name of the model database, and then click **Next** to continue.



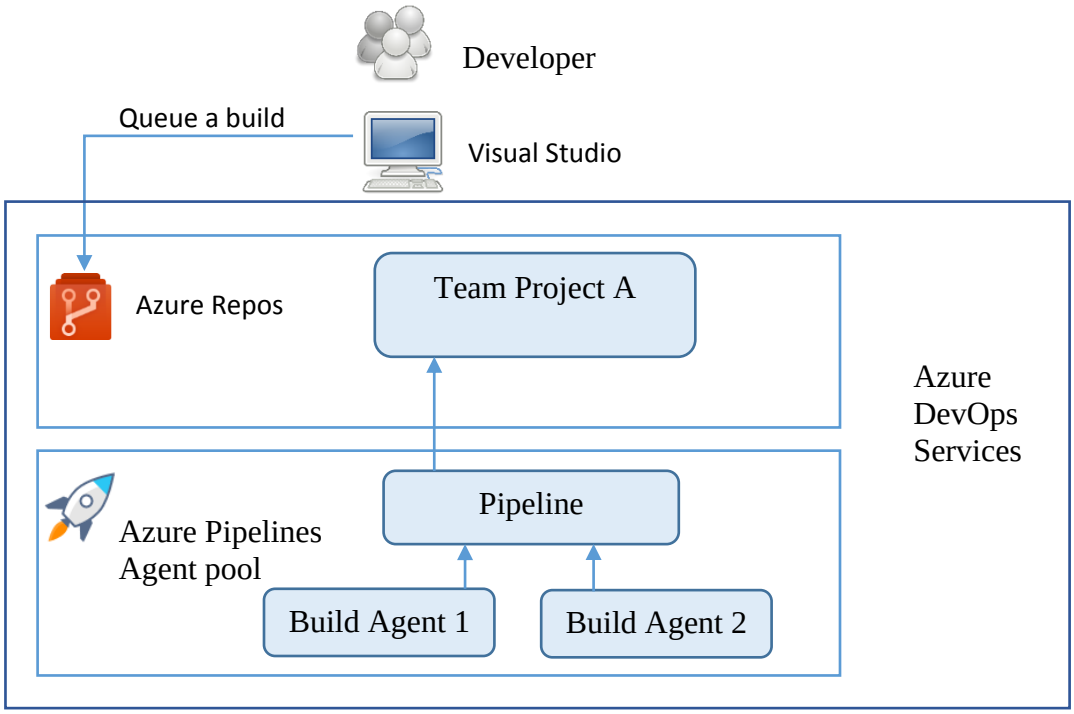
   If the model size is large, creating a new model database is going to consume time. You can clone an existing model database and then connect it by selecting **Existing**.

   Refer to *How to Clone an AB Suite Model Database* for more information.
7. Click **Finish** after it completes the process.

   Your own model database is created successfully and you can perform the check in or check out operations from the **Solution Explorer** or the **Class View** now.

Building an AB Suite Model using Azure Pipelines

Azure Pipelines helps you to implement a build, test and deployment pipeline for AB Suite application. To build or deploy AB Suite application using Azure Pipelines you need at least one agent. As you add more code and users, you will eventually need more agents.



The preceding diagram is a typical set up on how to build an Azure Pipelines and recommended for AB Suite. Each pipeline agent that hosts the Build Agents requires a full install of the AB Suite Developer and SQL Server software. This part will guide you through the required settings to allow Visual Studio build definitions to work with AB Suite projects. The build agent can be installed on a local machine or an Azure VM.

Configure the Build Agent

On the build agent, you need to:

- Install AB Suite Developer
- Install SQL Server to host the AB Suite model database
- Run the build service as an administrator
- Install self-hosted agent

Regarding the self-hosted agent installation instructions, refer to Appendix A.

Note: *To install self-hosted agent in a machine, the user account should be an administrator for the Agent pool you plan to install. For example, an Azure DevOps Services organization owner is an administrator. If you want to use another user*

account to install agent, you should ask Azure DevOps Services organization owner to add it as an administrator.

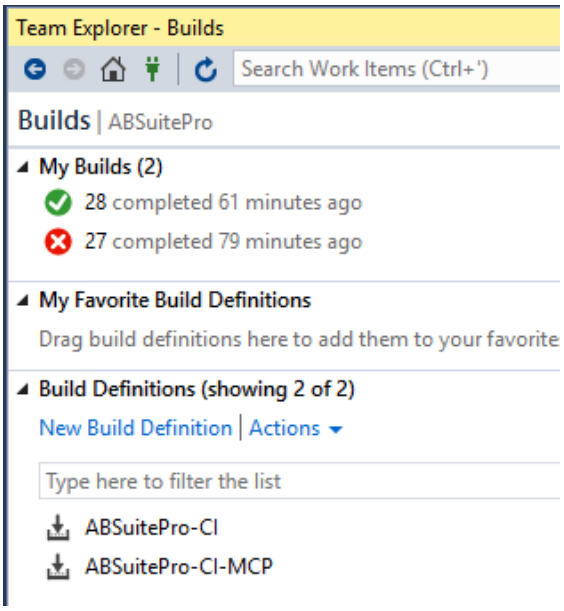
Configure a Build Definition

To build an AB Suite application from Azure DevOps Services, you must create a build definition. Open Visual Studio to create a Visual Studio build definition and ensure the **Solution** to build, **Configuration**, **Platform**, **MSBuild Arguments**, and **MSBuild Architecture**, **Visual Studio Version** are configured correctly. This document only covers the additional settings for AB Suite. For settings related to **Build Pipeline trigger**, refer to Appendix A.

Create a Build Definition

1. In the **Team Explorer**, click **Builds**.

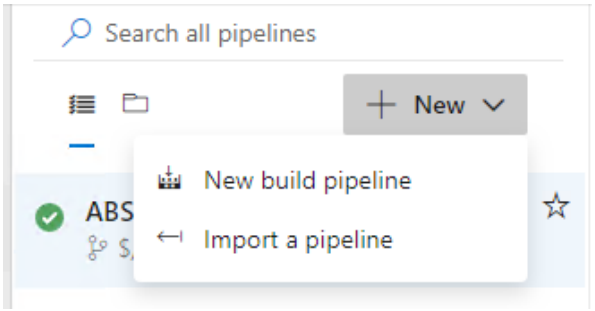
   The following window appears.



2. Click **New Build Definition**.

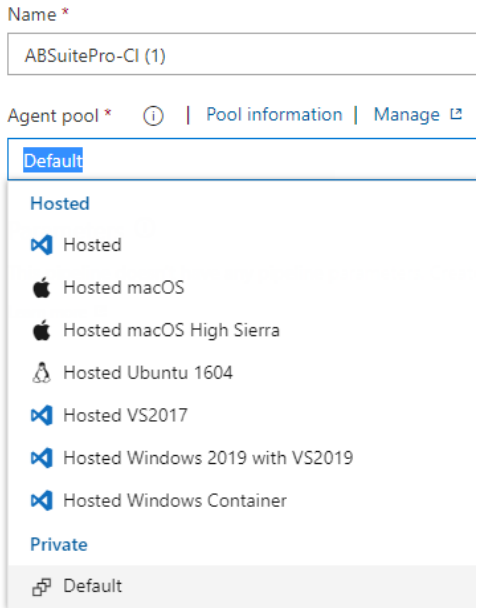
   The build definition website opens.

3. Create a **New build pipeline**.

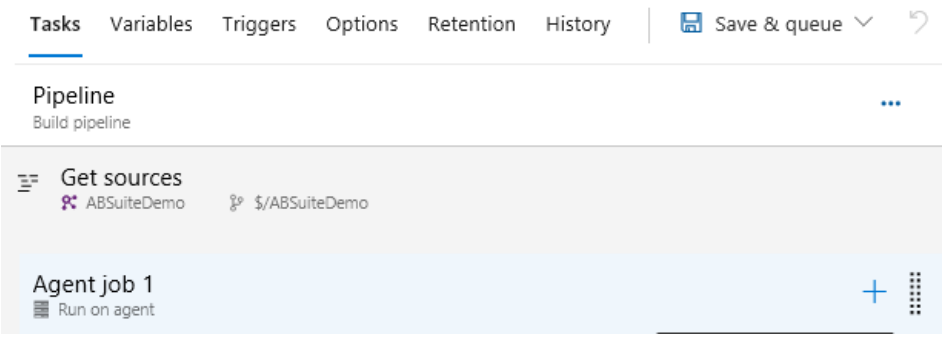


4. Start with an **Empty pipeline** template.

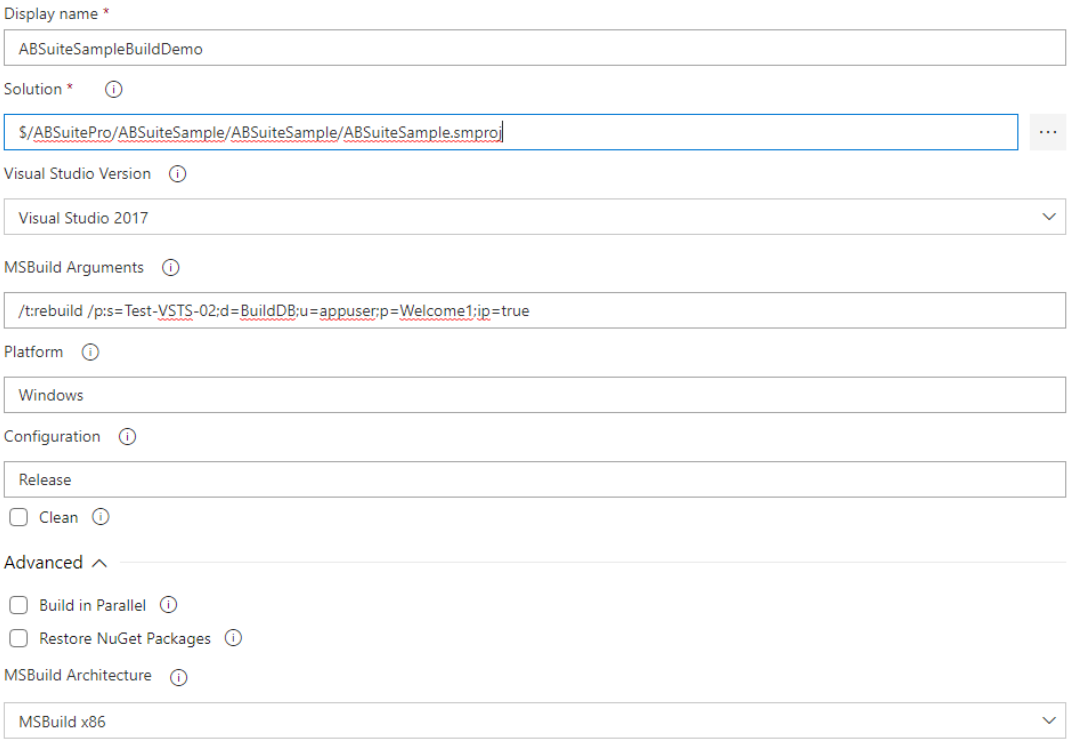
5. Specify a name for the pipeline and select the private **Agent pool** you configured while installing self-hosted agent (for example, Default).



6. Click **+** icon to add a **Visual Studio build** task to the agent.



7. Specify the details of these boxes namely **Display name**, **Solution**, **Visual Studio Version**, **MSBuild Arguments**, **Platform**, **Configuration** and **MSBuild Architecture**.



Build Process Parameters

In the process section of a build definition, you need to provide values for the **Solution**, **Platform**, **Configuration**, **MSBuild Arguments** and **MSBuild Architecture**.

Solution: AB Suite supports System Modeler project files (.smproj). Select the project through the button "…".

Platform: Specify the platform on which the application is built.

Configurations: Specify the configuration name for the AB Suite project that you are building.

MSBuild Arguments: Enter the build arguments in the MSBuild arguments field using the following format:

```
/t:<buildmode>
/p:s=<build_sqlserver>;d=<model_database>;u=<host_usercode>;p=<ho
st_password>;fu=<ftpusercode>;fp=<ftppassword>;ip=true
```

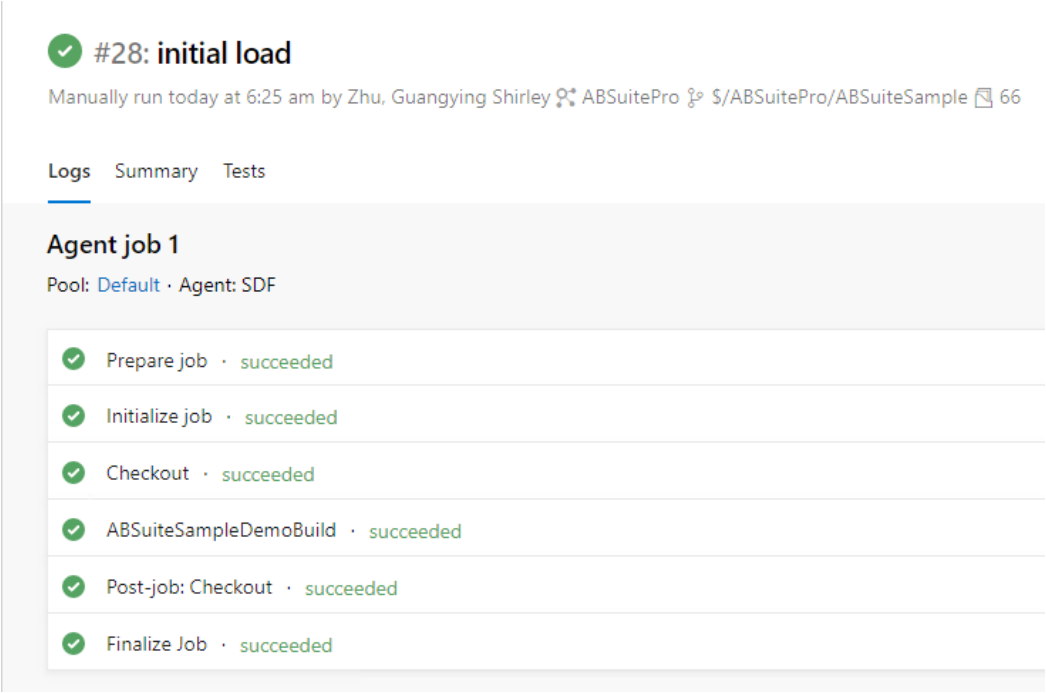
Note: *The <model_database> specified in the MSBuild arguments should be a separate database from that being used for development. For performance reasons, it is recommended that this model database reside in an instance of SQL Server installed on the build server. Delete the MCP related properties (fu and fp) when building the application for Windows.*

The following table provides details of the most common switches and properties. For a full set of MSbuild properties, refer to the *Agile Business Suite Developer Online Help* for more information.

Switch/ Property	Name	Description
/t:buildmode	Target	Build mode: rebuild, build, or clean.
/p: name=value	Property	Properties to build AB Suite application, where name is the property name and value is the property value. Specify each property separately, or use a semicolon to separate multiple properties. For example, /p:pf=windows;c=release.
ac	AccessCode	Access code for an MCP host.
ap	AccessPassword	Access password for an MCP host.
c	Configuration	Name of a configuration that you want to build.
cc	ChargeCode	Charge code for an MCP host.
d	Database	Name of a model containing the application you want to build.
fp	FTPPassword	FTP password for an MCP host.
fu	FTPUserCode	FTP user code for an MCP host.
ip	ImportProject	The project will be imported and a new model database (specified by the property d) will be created, if the database does not exist when this property is set to true.
p	Password	Password used to connect to the host machine.
pf	Platform	Name of a platform on which the application is built.
s	SQLServer	SQL Server that contains the AB Suite model to build. "(Local)" is the default local SQL Server instance.
u	UserCode	User code to connect to the host machine.

MSBuild Architecture: Choose MSBuild x86 to always process this build using the 32-bit version of MSBuild.

You have now completed the configuration for the build definition for building an AB Suite application. When a build is triggered, (for example, queue the build manually), you can check the build status in the Team Explorer and see the detail results by clicking the status, as follows:



Appendix A — Related Articles

- Agile Business Suite Developer User Guide
- Quick start on using Azure DevOps Services

  https://docs.microsoft.com/en-us/azure/devops/user-guide/index?view=azure-devops
- Azure DevOps Services Documentation

  https://docs.microsoft.com/en-us/azure/DevOps/?view=azure-DevOps
- Azure virtual machines documentation

  https://docs.microsoft.com/en-us/azure/virtual-machines/windows/
- Estimate the costs for Azure virtual machines

  https://azure.microsoft.com/en-au/pricing/calculator/
- Enable Preview Features

  https://docs.microsoft.com/en-us/azure/devops/project/navigation/preview-features?view=azure-devops
- Set permissions at the project or collection level

  https://docs.microsoft.com/en-us/azure/devops/organizations/security/set-project-collection-level-permissions?view=azure-devops
- Self-hosted agent installation instructions

  https://docs.microsoft.com/en-us/azure/devops/pipelines/agents/v2-windows?view=azure-devops
- How to Clone an AB Suite Model Database

https://www.support.unisys.com/abs/docs/Howto/How%20to%20Clone%20an%20AB%20Suite%20Model%20Database.zip

- Visual Studio Build Task

  https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/build/visual-studio-build?view=azure-devops

- Build Pipeline triggers

  https://docs.microsoft.com/en-us/azure/devops/pipelines/build/triggers?view=azure-devops&tabs=classic

For more information visit www.unisys.com

© 2022 Unisys Corporation.
All rights reserved.