UNISYS

Agile Business Suite

How To: Team Foundation Build DevOps Support for Windows

Applies to: AB Suite 6.1

December 2016

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.

Team Foundation Build DevOps Support for Windows

Summary

Agile Business Suite 6.1 (AB Suite™) Runtime for Windows now provides integration with Team Foundation Build. With this feature, you can automate the Build process from one Model to different environments, such as test, acceptance, and production by leveraging Team Foundation Build.

As part of the DevOps process improvements, AB Suite 6.1 Runtime for Windows now allows the location information (for example, Extract File Location) for different environments to be set post-generate, without building the model again.

You can deploy either online system or deploy reports only by using Team Foundation Build and reconfigure them without regeneration. This document provides the instructions on how to do it. You need to:

- Setup Team Foundation Build DevOps Environment
- Create Your DevOps Scripts
- Define Your Build Process

Additionally, a sample project with the model files and the PowerShell script is provided along with this document.

This document is applicable for AB Suite 6.1 onwards. AB Suite 6.1 Developer using Visual Studio 2015 Professional Edition Update3 from MSDN Subscription is used for the screenshots in this document.

1. Setup Team Foundation Build DevOps Environment

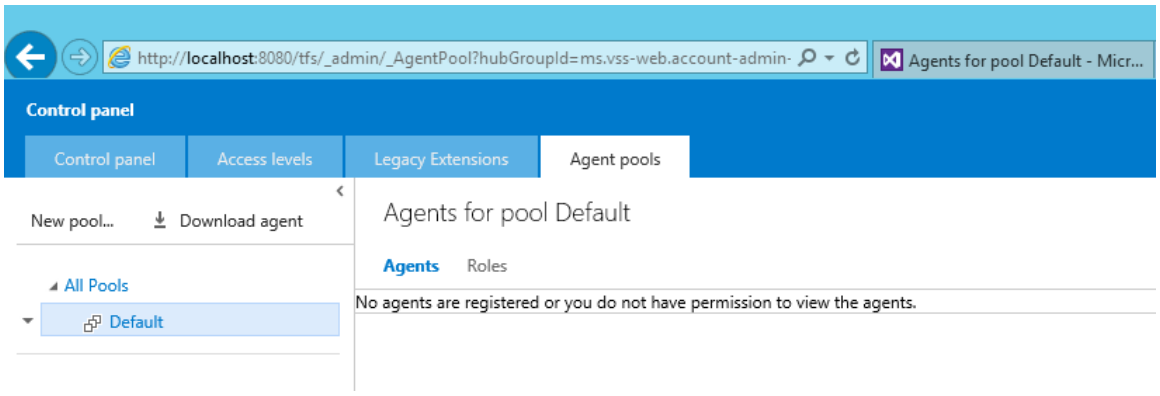
To build your code or deploy your software you need at least one agent, and as you add more code and people, you'll eventually need more. In this document, only one build agent is installed on the same server where Team Foundation Server is installed.

You need to install the following software, check Agile Business Suite Qualification Matrix for more information:

- Microsoft SQL Server
- Visual Studio 2015
- Agile Business Suite Developer
- Agile Business Suite Windows Runtime
- Team Foundation Server 2015
- Team Foundation Server Build Agent

You can use the default settings when installing Team Foundation Server 2015. There is an option to allow you to install the build agent when installing Team Foundation Server 2015. If you do not select that option, you can also follow the steps below to install the build agent.

A. On the machine where you are going to run the build agent, open a web browser and go to the Agent pools control panel tab.



B. Click **Download agent**.

C. Unzip the .zip file into the folder on disk from which you would like to run the agent. To avoid "path too long" issues on the file system, keep the path short. For example: C:\Agent\

D. Run Command Prompt as Administrator, and then run: C:\Agent\ConfigureAgent.cmd

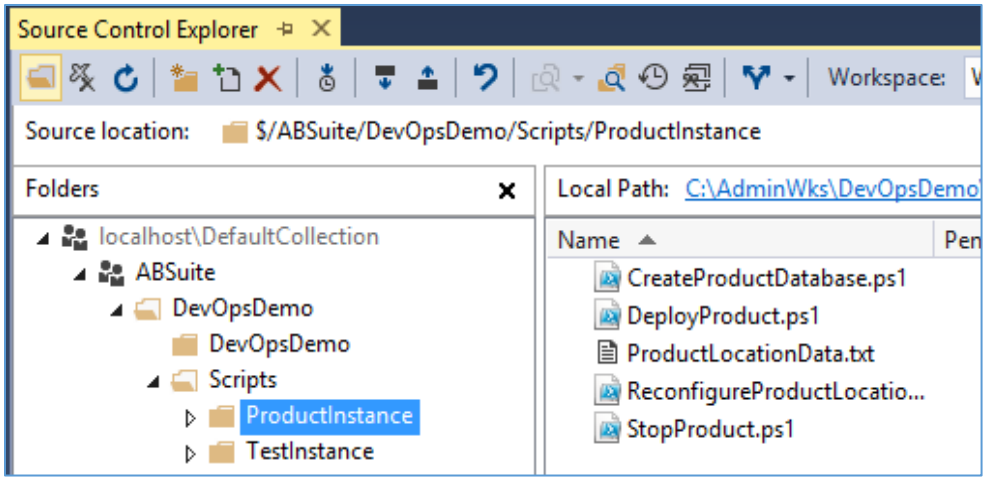
E. Respond to the prompts.

   Enter the **Administrator** user account when it ask the user account for running the build agent services. Otherwise the build may fail due to lack of permission.

Note: *On the machine where the build agent is installed, make sure you login as the user who run the build agent service and then open the Visual Studio, or you can run the script in Appendix A.*

2. Create Your DevOps Scripts

As part of the DevOps process, you may want to stop the system, create the database, clone an existing application MSI, and deploy the cloned application MSI. You can create a script (for example, a PowerShell script) for each activity and add those scripts to Team Foundation Server (TFS). These scripts can be used in the build definition. The following is a screenshot of the scripts checked into TFS, all of them are put under the Scripts folder. See below:



2.1. Stop System

You can use the AdminSystem.exe to stop the existing system before you deploy it again. It is always recommended to stop the system before you deploy the system again. Here is one example:

"C:\Program Files\Unisys\AB Suite 6.1\Bin64\AdminSystem.exe" /S DevOpsTest /STO

For more information about the AdminSystem.exe, refer to the Agile Business Suite Runtime for Windows Operating System Administration Guide.

2.2. Clone Application MSI and Reconfigure Locations

You can use the Deployment Package Manager Utility (ManagePackage.exe) to clone an AB Suite application MSI and reconfigure the location objects. This is an optional step, you do not have to always clone the application MSI and reconfigure locations. See one example below:

"C:\Program Files\Unisys\AB Suite 6.1\Bin64\ManagePackage.exe" /C /BP C:\intermediate\Sample.msi /TP C:\intermediate\Clone /DB DevOpsTestDB /DBS DevOpsTest /SN DevOpsTest /PN DevOpsTest /CFD "C:\DevOps\ReconfigLocation.txt" /O true

The parameter /CFD is new parameter introduced in 6.1. It allows to specify a text file which have multiple lines. Each line are separated by the object name and the desired location. You can reconfigure the LOG Pack Location, ROC Pack Location, EXTRACT Pack Location, and System Pack Location.

For example, the C:\DevOps\ReconfigLocation.txt can be like this:

Sample.AuditLog C:\myauditlog

The information of the new parameter /CFD is not available in the Agile Business Suite Runtime for Windows Operating System Administration Guide yet. All the other parameters of ManagePackage.exe are available in the Agile Business Suite Runtime for Windows Operating System Administration Guide.

Note: *This Deployment Package Manager Utility is automatically installed with Agile Business Suite Runtime and is located in the AB Suite 6.1 installation directory. Refer to Appendix B for the Prerequisites for DPM for more information about the prerequisite information.*

2.3. Create Database

The database must be created before you deploy an application in a clean environment. Database Configuration commands (creating and deleting a runtime database, attaching an existing database, adding and deleting database server registrations, attaching an existing database) now added to Runtime API. This can be accessed through the new ConfigureDatabase.exe Runtime API utility, or programmatically through IConfigureDatabase. This is an optional step since the database may have already been created manually.

Here is one example of registering a database server and creating a new database:

"C:\Program Files\Unisys\AB Suite 6.1\Bin64\ConfigureDatabase.exe" /REG /ALS default

"C:\Program Files\Unisys\AB Suite 6.1\Bin64\ConfigureDatabase.exe" /C /DBN DevOpsTestDB /ALS default

Note: *The information of the ConfigureDatabase and is IConfigureDatabase is available in the Agile Business Suite Runtime for Windows Operating System Administration Guide, you can also refer to the Appendix C for more information.*

2.4. Deploy Cloned Instance

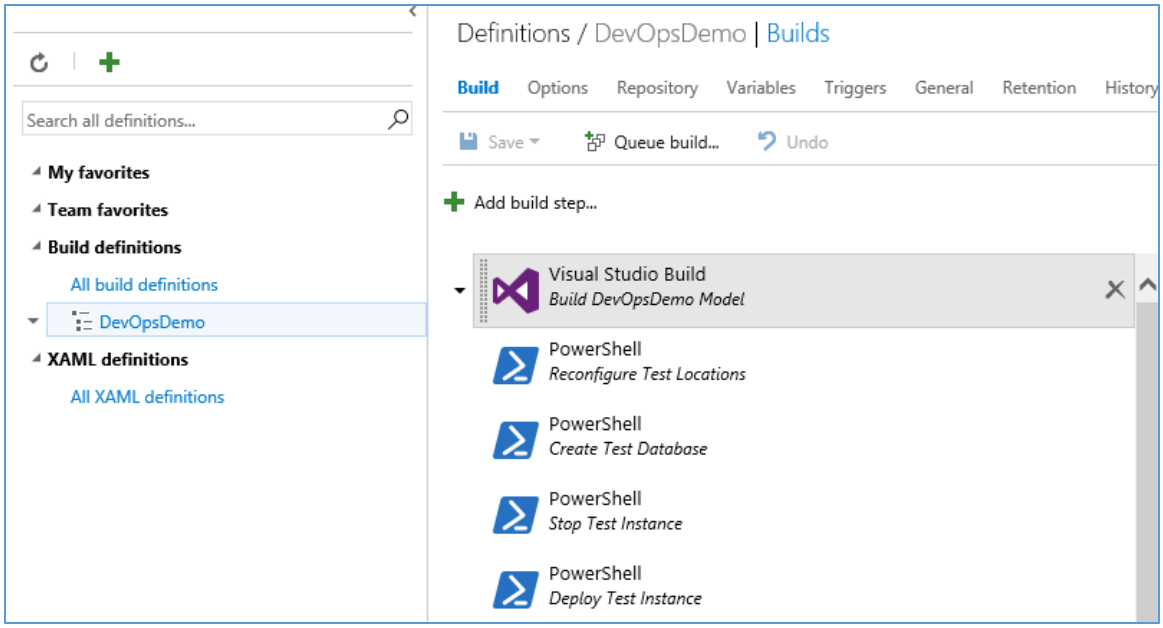
When everything is ready, you can use the AB Suite System Deployment Utility (DeployPackage.exe) to deploy a cloned application MSI to a local or remote server. See the example below:

"C:\Program Files\Unisys\AB Suite 6.1\Bin64\DeployPackage.exe" /L "C:\intermediate\Clone\DevOpsTest.msi" /U AppUser /P Password

For more information of the DeployPackage.exe refer to the Agile Business Suite Runtime for Windows Operating System Administration Guide.

3. Define Your Build Process

With Team Foundation Build, you can define your build process to build an AB Suite application, clone the generated application MSI and deploy it for different purpose (Testing, Production, etc.). Here is the screenshot of one example of the build process:

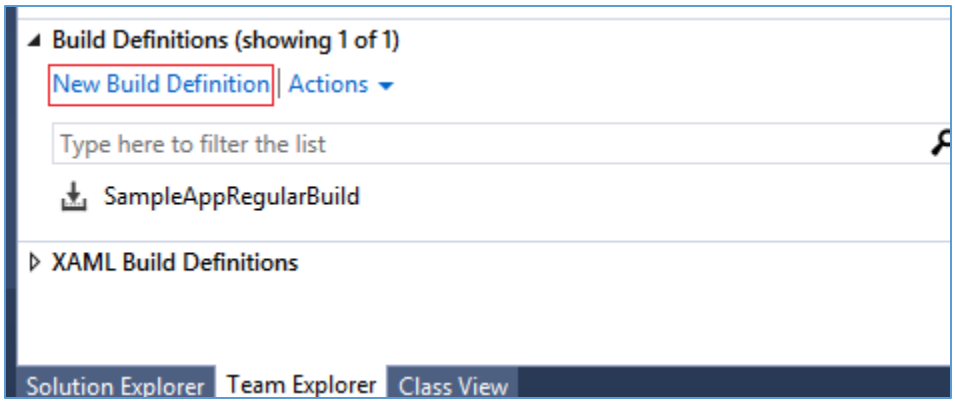


The above example includes several build steps:

- Build AB Suite Application
- Clone Application MSI and Reconfigure Locations
- Create Database
- Stop System
- Deploy Cloned Instance

3.1. Create a New Build Definition

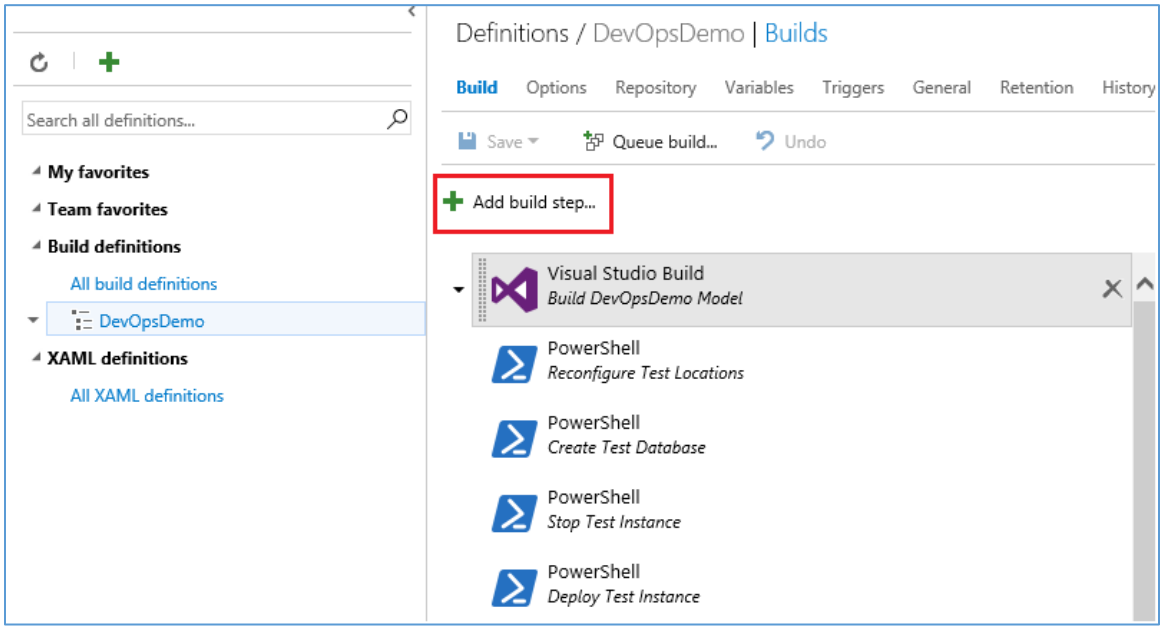
Adds AB Suite Project to TFS before you start to define your build process. To automate the Build process with Team Foundation Build, you need to create a build definition. A new build definition can be created from Visual Studio.



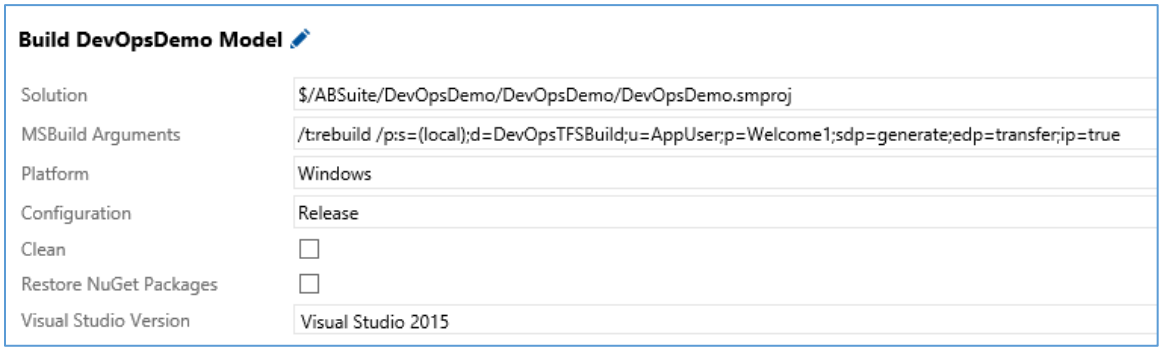
See the screenshot above. In Team Explorer, click the **New Build Definition** button and then start with an empty template, then add Visual Studio Build, Batch Script, PowerShell script, etc to the build definition.

3.2. Add a Visual Studio Build Build Step

To build AB Suite application with Visual Studio, you need to add **Visual Studio Build** build step to your build definition. Click **Add Build Step** to add a new build step.



Make sure the **Solution**, **MSBuild Arguments**, **Platform**, **Configuration, Restore NuGet Packages**, and **MSBuild Architecture** are configured correctly. This document only covers the additional settings for AB Suite. For the details of other settings, refer to Appendix D.



Solution: AB Suite supports System Modeler project files (.smproj). Open the SELECT PATH dialog box and select the AB Suite project file you are building.

MSBuild Arguments: Enter MSBuild arguments using the following format:

/t:*<buildmode>*
/p:*s=<build_sqlserver>;d=<model_database>;u=<host_usercode>;p=<host_pass word>;sdp=<start_deploy_phase>;edp=<end_deploy_phase>;ip=true*

If you want to build a report only, you can create a build definition and specify the report that you want to build with the element list parameter (el) in the MSBuild Arguments.

For example:
/t:rebuild
/p:s=(local);d=SampleApp;u=AppUser;p=Welcome1;sdp=generate;edp=transfer;el =Sample.Reports.CustList;ip=true

The following table provides the details of the most common switches and properties. For a full set of MS build properties refer to the AB Suite Online Help.

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
c	Configuration	Name of a configuration that you want to build.
d	Database	Name of a model containing the application you want to build.
edp	EndDeployPhase	Termination of the deploy phase for a build.
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.

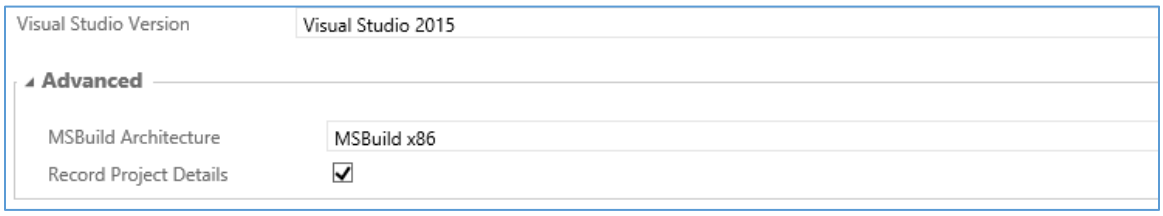
s	SQLServer	SQL Server that contains the AB Suite model to build. "(Local)" is the default local SQL Server instance.
sdp	StartDeployPhase	Initiation of the deploy phase for a build.
u	UserCode	User code to connect to the host machine.

Platform: DevOps only support Windows platform. Type Windows here.

Configuration: Type the Configuration for the AB Suite project that you are building. E.g., Release.

Restore NuGet Packages: Uncheck this option since AB Suite do not require this option enabled.

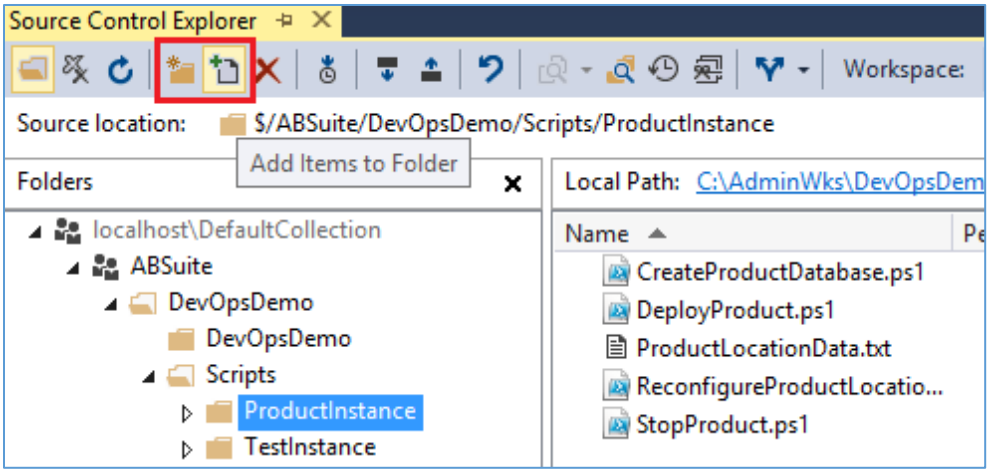
MSBuild Architecture: Specify MSBuild x86 to always process this build using the 32-bit version of MSBuild.



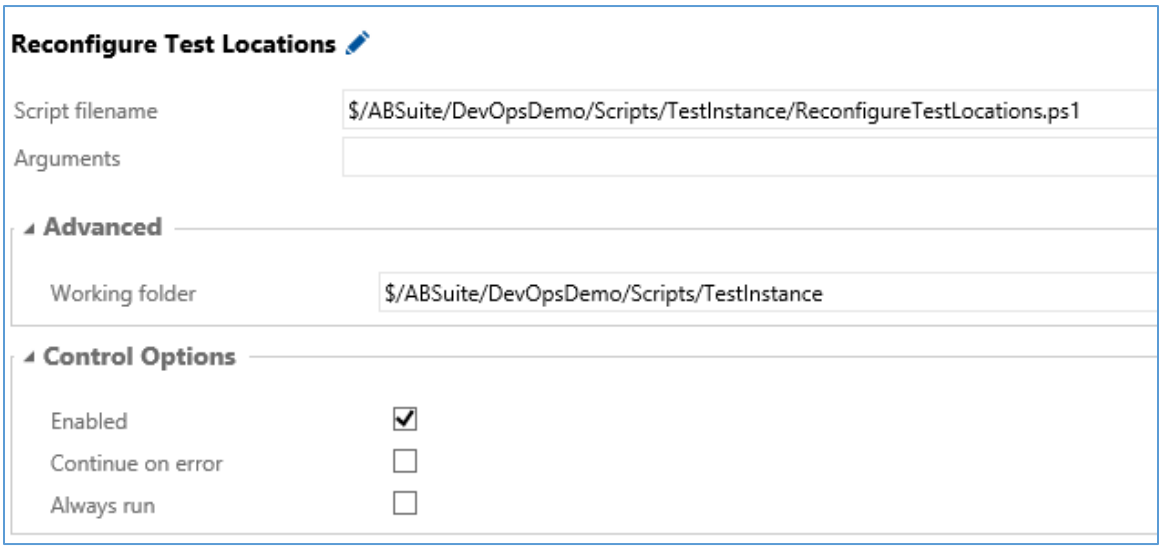
3.3. Add PowerShell Script Build Steps

Team Foundation Build does not know how to stop an AB Suite application, it does not know how to deploy an AB Suite application either, but it allows you to add PowerShell Script build step to your build definition so that you can use a PowerShell script to do that.

Make sure the PowerShell script files (the script files created in Create Your DevOps Scripts section) are checked-in into Team Foundation Server before you add the PowerShell Script build step. You can add a folder (for example, Scripts folder) by clicking the New Folder button and add the batch script file under the folder. Check in the script file after you add the PowerShell script file under the new created folder. See below:

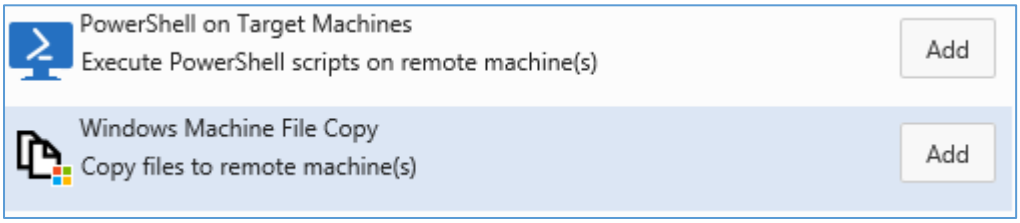


Now you can add several PowerShell Script build steps for the scripts that you created in the *Create Your DevOps* Scripts section and specify the script path for each PowerShell Script build step. See one example below:

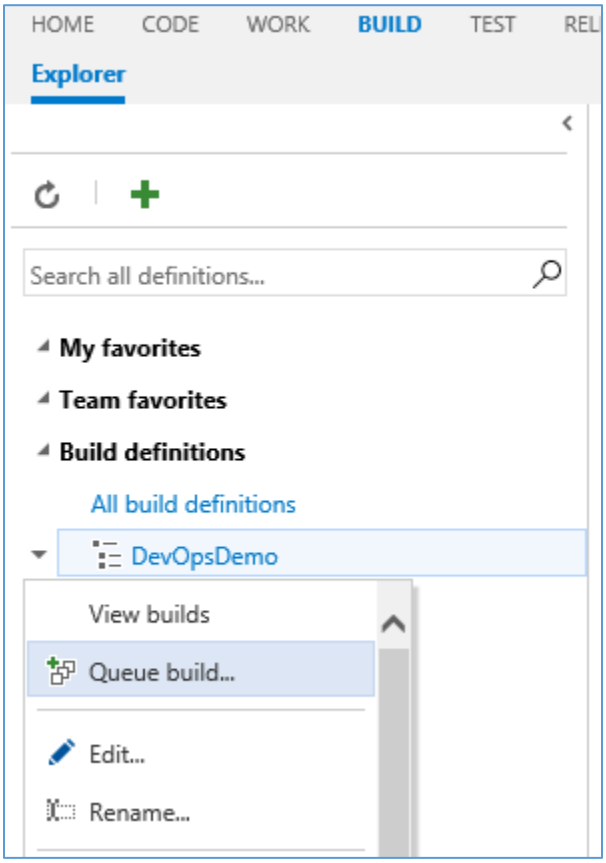


You can also drag each build step to change the sequence of build step. Save the build definition after you complete adding all your build steps.

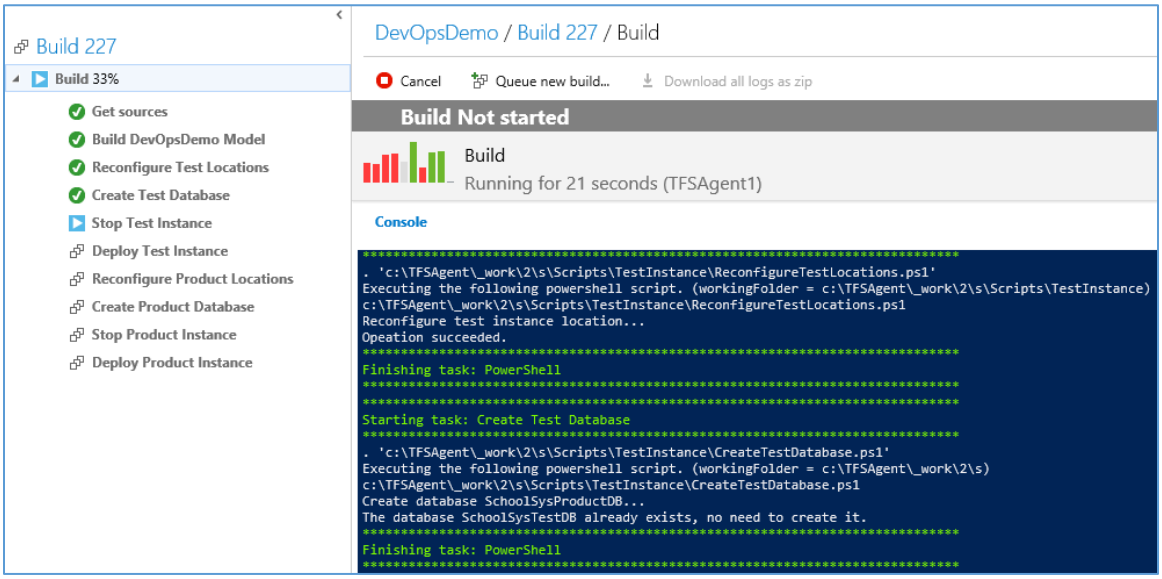
Note: *If you want to deploy the cloned application MSI to a remote server, you can add build steps to copy the files to remotes machine and add build steps to execute PowerShell scripts on remote machines.*



Now everything is ready, you can queue build from web portal or Visual Studio. See below:



It will build your AB Suite application with Visual Studio and run the script to deploy the AB Suite application when the build starts. You can see the progress in the console. See below:



You can check whether the cloned application MSI is deployed properly after the build is completed.

Known Issues/Limitations

1. The location object will be reset to its original value when deploying the cloned partial Deployment Package. The workaround is to reconfigure the location object when cloning the partial Deployment Package.

2. The updated location of the original report and that of the cloned report will be the same when a cloned report is deployed to the same system with a different name.

Appendix A

Script to open visual studio

Put the following script in a batch file and then run it.

```
echo off
REM ***************************************
REM * Open Visual Studio *
REM ***************************************
REM ***************************************
REM * Parameters                  *
REM ***************************************
set /p user=Enter the name of the user account to use for the build agent service:
echo user:  %user%

runas  /user:%user%  "%VS140COMNTOOLS%..\IDE\devenv.exe"
```

Appendix B

Prerequisites for DPM

For DPM to work, the following applications must be present in your computer.

• **CSC.exe** – C# Compiler is required for compiling and generating new COM+ binary. Microsoft .NET framework should be installed for installing CSC.exe. After Installation, the CSC.exe is available at: **C:\Windows\Microsoft.NET\Framework64\v4.0.30319** in the machine.

Note: *If you do the cloning and reset location information by executing DPM from Command Prompt (DPM-Console.exe), then you need to set the path of the CSC.exe to C:\Windows\Microsoft.NET\Framework\v4.0.30319\.*

• **SN.exe** – Strong Name (SN) generator is required for generating strong name for new COM+ application assembly. Microsoft SDK should be installed for installing SN.exe. After installation, the SN.exe is available at: **C:\Program Files (x86)\Microsoft SDKs\Windows\v8.1A\bin\NETFX 4.5.1 Tools** in the machine.

• **MSIDB.exe** – MSIDB.exe is required for analyzing input MSI. Microsoft SDK should be installed for installing MSIDB.exe. After installation, the MSIDB.exe is available at: **C:\Program Files (x86)\Windows Kits\8.1\bin\x86** in the machine. For some of the OS, you may find the MSIDB.exe is available at **C:\Program Files\Microsoft SDKs\Windows\v7.0\Bin** or **C:\Program Files\Microsoft SDKs\Windows\v7.0A\bin**.

• CABARC.exe – CABARC.exe is required for creating cabinet file for MSI. CABARC.exe is available at: C:\Program Files\Unisys\AB Suite 6.1\Bin64

The DPMSettings.xml file contains the path of all the pre-requisite applications. This file is automatically created when Deployment Package Manager is run for the first time. This file is automatically created at **C:\ProgramData\Unisys\ABSuite\6.1\DPM**.
When this file is created for the first time, DPM automatically locates the path of all pre-requisite applications. We can also edit the paths in this file as per our need.

Appendix C

Configure Database Utility (ConfigureDatabase.exe)

The Configure Database utility allows you to perform database configuration operations. You can perform the operations either by executing the ConfigureDatabase.exe or by using the Runtime Administration Tool. Following are the operations:

- Create Database: Databases can be created and configured by executing the ConfigureDatabase.exe.

- Database Server Registration: Once you select a server machine, you need to add a database server registration for each of the runtime servers to which you want to add databases. You can add a database server registration by executing the ConfigureDatabase.exe.

- Attach an existing Database: To add an existing database, you must attach it to the new Windows Runtime environment by executing the ConfigureDatabase.exe.

- Detach a Database: Detaching a database involves the removal of database from the runtime environment but does not delete the database or the data it contains. You can detach the database by executing the ConfigureDatabase.exe.

- Delete a Database: Deleting a database involves the complete removal of the database from the server. You can only delete a database if it does not contain any applications. You can delete the database by executing the ConfigureDatabase.exe.

- Removing a Database Server Registration: Database Server Registration can be removed by executing the ConfigureDatabase.exe.

This utility is automatically installed with Agile Business Suite Runtime and is located in the AB Suite 6.1 installation directory. You can invoke the utility from the command line to configure database.

Note: *By default, only a local administrator can use the ConfigureDatabase.exe.*

To run ConfigureDatabase.exe from a command line:

1. Point to the bottom-left corner of the screen to enable the **Start** icon, right-click **Start**, and then click **Run**.
2. Type cmd in the **Run** dialog box and press **Enter** to open a command prompt.
3. In the command prompt, change the working directory to the /bin folder of the AB Suite 6.1 installation directory.
   cd C:\Program Files\Unisys\AB Suite 6.1\Bin64
4. Type "ConfigureDatabase.exe /?". This displays the complete usage of the utility including the parameters.

The syntax of ConfigureDatabase for creating a database is:

ConfigureDatabase /C /DBN <database name> /ALS <database server alias> [/?]

Parameters

Argument	**Description**
/C	Creates a new database on the database server.
/DBN	Specifies the name of the database that will be created. ***Note:*** *This argument is mandatory.*
/ALS	Specifies the database server registration under which this database is created. ***Notes:*** • *This argument is mandatory.* • *If the database server registration does not exist, it will be automatically created as part of the process of creating the database and associated with the default instance of SQL server on the runtime server.*
/?	Displays the command line syntax and arguments of the utility. If specified, a usage message appears and the utility exits.

The syntax of ConfigureDatabase for deleting a database is

ConfigureDatabase /D /DBN <database name> /ALS <database server alias> [/?]

Parameters

Argument	**Description**
/D	Deletes the specified database from the database server.

/DBN	Specifies the name of the database that will be deleted. ***Note:*** *This argument is mandatory.*
/ALS	Specifies the database server registration under which the database that will be deleted exists. ***Notes:*** • *This argument is mandatory.* • *The database will be deleted only if it is not in use by applications.*
/?	Displays the command line syntax and arguments of the utility. If specified, a usage message appears and the utility exits.

The syntax of ConfigureDatabase for attaching a database is

ConfigureDatabase /AT /DBN <database name> /ALS <database server registration> [/?]

Parameters

Argument	**Description**
/AT	Attaches the specified database to the runtime environment.
/DBN	Specifies the name of the database that will be attached. ***Note:*** *This argument is mandatory.*
/ALS	Specifies the database server registration under which the database will be attached. ***Note:*** *This argument is mandatory*
/?	Displays the command line syntax and arguments of the utility. If specified, a usage message appears and the utility exits.

The syntax of ConfigureDatabase for detaching a database is

ConfigureDatabase /DT /DBN <database name> /ALS <database server registration> [/?]

Parameters

Argument	**Description**
/DT	Detaches the specified database from the runtime environment.
/DBN	Specifies the name of the database that will be detached. ***Note:*** *This argument is mandatory.*
/ALS	Specifies the database server registration from which the database will be detached. ***Note:*** *This argument is mandatory*
/?	Displays the command line syntax and arguments of the utility. If specified, a usage message appears and the utility exits.

The syntax of ConfigureDatabase for creating a database server registration is

ConfigureDatabase /REG /ALS <database server alias> [/SQL <sql instance name>] [/H <database server machine>]

Parameters

Argument	**Description**
/REG	Creates a database server registration for the runtime environment.
/ALS	Specifies the database server registration that will be created. ***Note:*** *This argument is mandatory.*
/SQL	Specifies the SQL instance name which the database server alias represents. If this parameter is not specified, the default instance is assumed.
/H	Specifies the name of the database server. If this parameter is not specified, localhost is assumed.

/?	Displays the command line syntax and arguments of the utility. If specified, a usage message appears and the utility exits.

The syntax of ConfigureDatabase for removing a database server registration is

ConfigureDatabase /URG /ALS <database server alias>

Parameters

Argument	**Description**
/URG	Removes a database server registration from the runtime environment.
/ALS	Specifies the database server registration that will be removed. ***Note:*** *This argument is mandatory.*
/?	Displays the command line syntax and arguments of the utility. If specified, a usage message appears and the utility exits without starting the deployment process.

Appendix D

1. Set up Team Services or Team Foundation Server:
   https://www.visualstudio.com/en-us/docs/setup-admin/get-started

2. Build for your platform, speaking your language
   https://www.visualstudio.com/en-us/docs/build/overview

3. Define your build process
   https://www.visualstudio.com/en-us/docs/build/define/create

4. Agile Business Suite Runtime for Windows Operating System Administration Guide
   http://www.support.unisys.com

5. Agile Business Suite Developer User Guide
   http://www.support.unisys.com

For more information visit www.unisys.com

© 2016 Unisys Corporation.
All rights reserved.