Hands-on Lab 5:
Adding New Stages to the Pipeline

Table of Contents

Objectives

This HOL demonstrates how to add new stages to the continuous delivery pipeline that you constructed in the previous labs.

Prerequisites

To complete this lab, you must have completed the implementation of the continuous delivery pipeline.

Time

You should be able to complete this lab in 30 minutes.

Exercise 1: Adding New Stages to the Pipeline

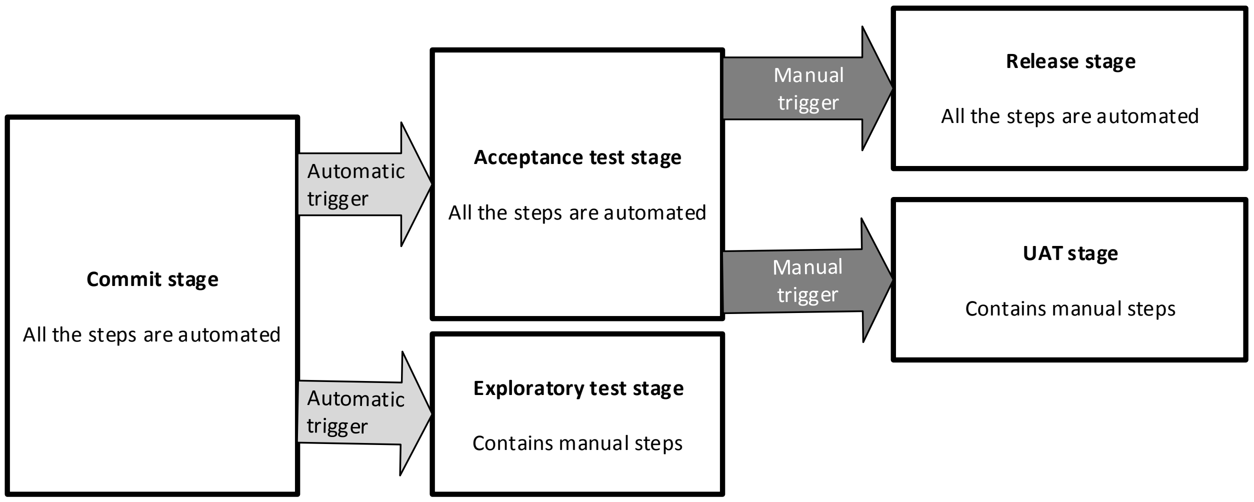
In this exercise , you learn how to add a new stage to the continuous delivery pipeline that you constructed in the previous labs. By adding new stages, you can improve the quality of your software. New stages allow you to add new validations to the release process.

Stages can be classified by their trigger and by whether the steps they contain are automated and/or manual. The four possibilities are shown in the following table.

Trigger	**Steps**	**Examples**
Automatic	Automated	The commit stage and the acceptance test stage in the continuous delivery pipeline you implemented.
Automatic	Automated and/or manual steps	An exploratory testing stage that is triggered automatically after the commit stage stage succeeds. The deployment step is done automatically but the exploratory testing is done manually.
Manual	Automated	A release stage that can be manually triggered when necessary. The deployment step is done automatically. This is the only step the stage contains.
Manual	Automated and/or manual steps	A UAT stage that can be manually triggered when necessary. The deployment is done automatically, after the stage is triggered. The UATs are manual.

In this exercise, you add an exploratory test stage, which is an example of an automatically triggered stage that contains both automated and manual steps. The stage automatically deploys the application to the target environment if the commit stage succeeds. After that, the testers complete the stage by manually performing exploratory tests.

This is how the pipeline will look after adding the Exploratory Test Stage:

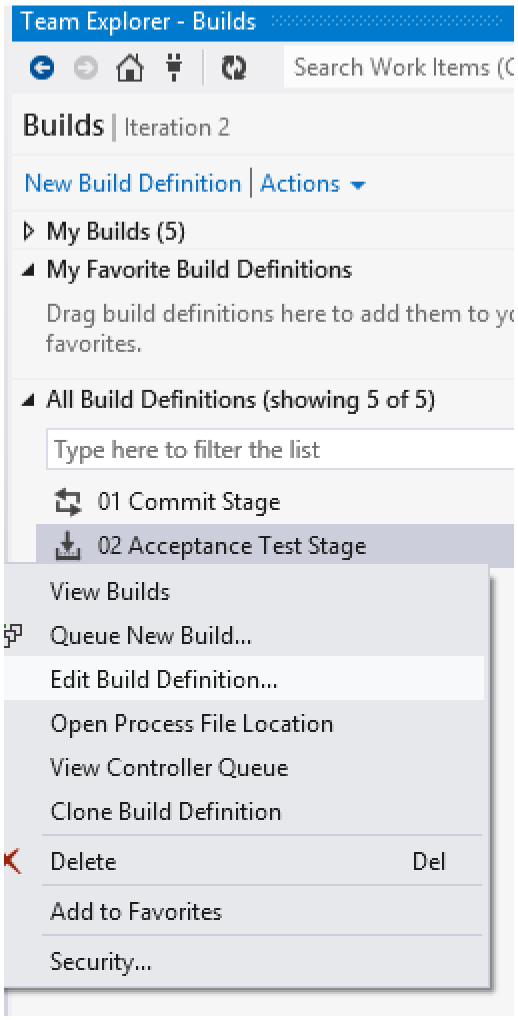


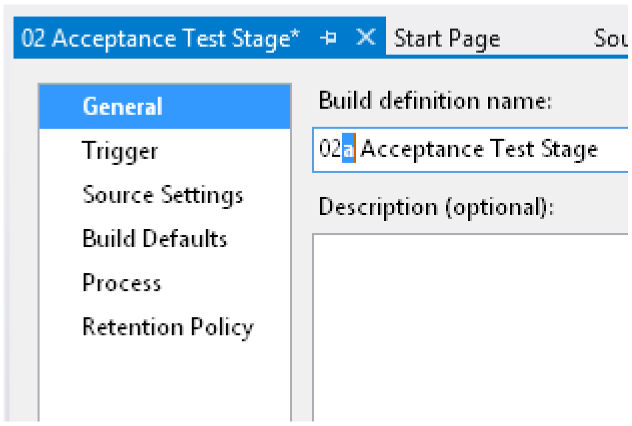
NOTE: For the sake of simplicity, this lab uses the same environments that were used in the previous labs. An improvement would be to add a new environment that is dedicated to exploratory testing. This would allow the pipeline to run multiple parallel stages until it reaches the release stage. Also, there would be fewer problems with timeouts that can occur when one stage is blocked because another is already using the environment.

Task 1: Create the Build Definition for the Exploratory Testing Stage

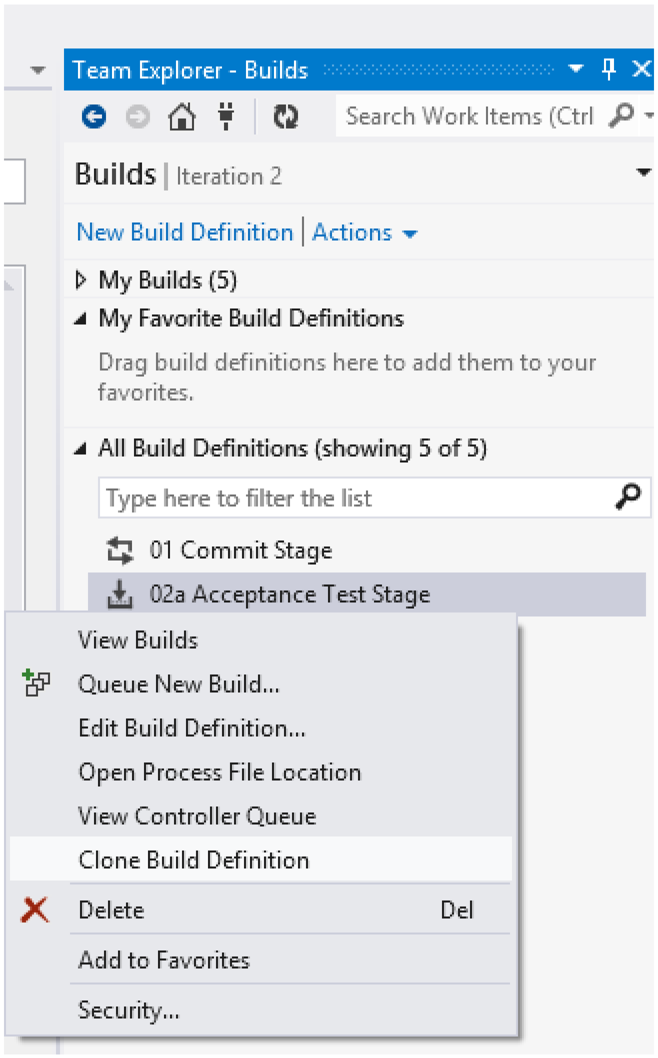
The exploratory testing stage is very similar to the acceptance test stage. It is automatically triggered and it performs an automated deployment using the same scripts and the same environment. The difference is that it has no automated tests, apart from BVTs that validate the deployment. Tests that validate the application are performed manually. Because of its similarity to the acceptance test stage that you've already implemented, you can base the exploratory testing stage on the **CDPipelineCommitStageProcessTemplate** that you customized in the orchestration HOL. In fact, you can use the acceptance test stage build definition as the basis of the new stage.

1. From **Team Explorer – Builds**, open the **02 Acceptance Test Stage** build definition. Right-click on it and select **Edit Build Definition**. Rename the build definition **02a Acceptance Test Stage**. Save the change.

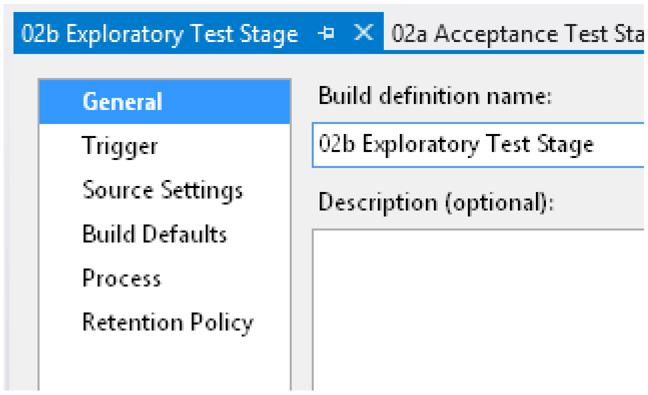




2. Clone the **02a Acceptance Test Stage** build definition.

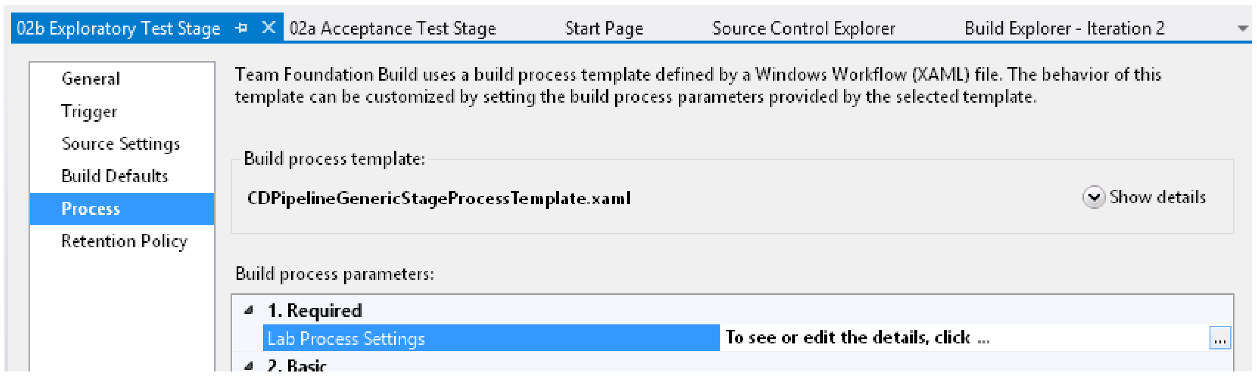

Team Explorer - Builds
Search Work Items (Ctrl
Builds | Iteration 2
New Build Definition | Actions
My Builds (5)
My Favorite Build Definitions
Drag build definitions here to add them to your favorites.
All Build Definitions (showing 5 of 5)
Type here to filter the list
01 Commit Stage
02a Acceptance Test Stage
View Builds
Queue New Build...
Edit Build Definition...
Open Process File Location
View Controller Queue
Clone Build Definition
Delete
Del
Add to Favorites
Security...

3. Change the name of the cloned build definition to **02b Exploratory Test Stage**.



Build definition name:
02b Exploratory Test Stage

4. Select the **Process** tab. Open the **Lab Process Settings** by clicking on the ellipsis **(…)**.


02b Exploratory Test Stage | 02a Acceptance Test Stage | Start Page | Source Control Explorer | Build Explorer - Iteration 2

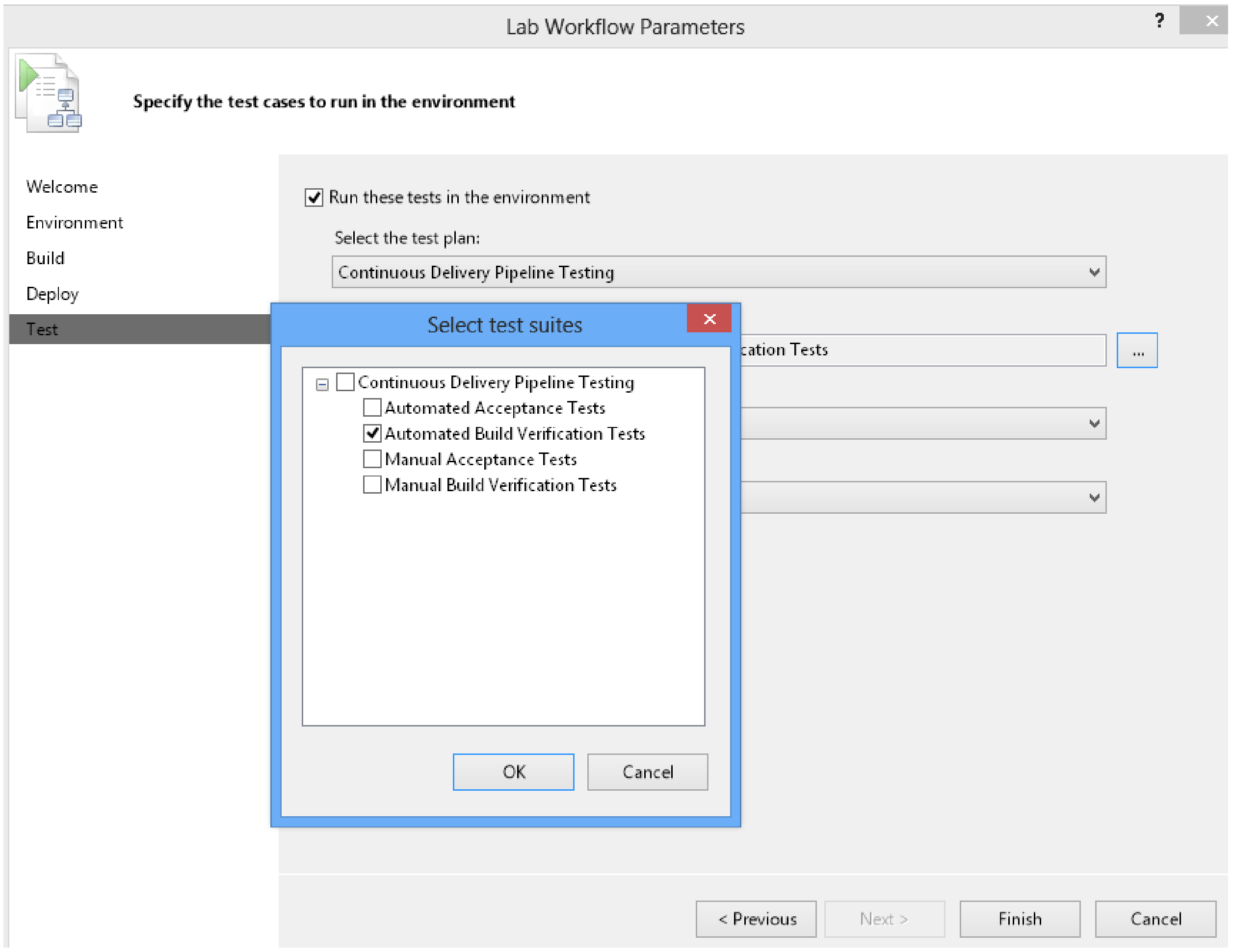
General
Trigger
Source Settings
Build Defaults
Process
Retention Policy

Team Foundation Build uses a build process template defined by a Windows Workflow (XAML) file. The behavior of this template can be customized by setting the build process parameters provided by the selected template.

Build process template:
CDPipelineGenericStageProcessTemplate.xaml
Show details

Build process parameters:
1. Required
Lab Process Settings — To see or edit the details, click ...
2. Basic

5. Select the **Test** tab. Leave **Automated Build Verification Tests** selected. Clear any other selections.


Lab Workflow Parameters

Specify the test cases to run in the environment

Welcome
Environment
Build
Deploy
Test

☑ Run these tests in the environment
Select the test plan:
Continuous Delivery Pipeline Testing

Select test suites
☐ Continuous Delivery Pipeline Testing
☐ Automated Acceptance Tests
☑ Automated Build Verification Tests
☐ Manual Acceptance Tests
☐ Manual Build Verification Tests

OK | Cancel

< Previous | Next > | Finish | Cancel

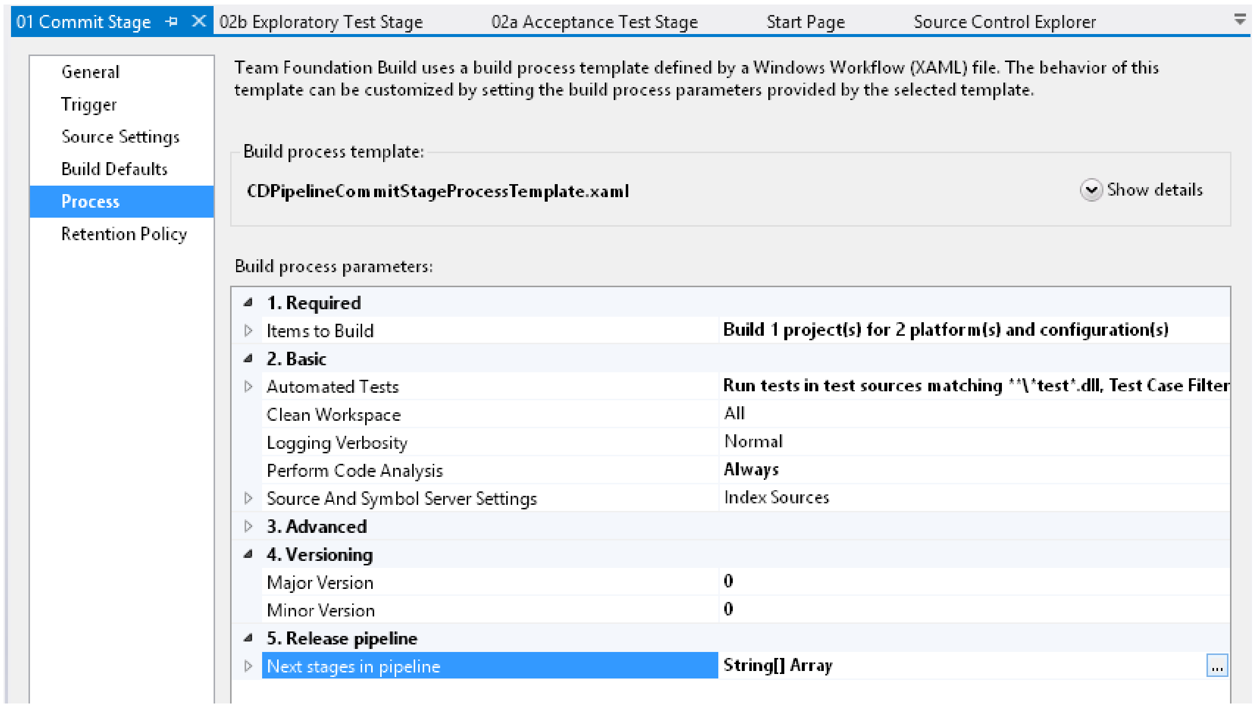
6. Click **Finish**. Save the build definition.

Task 2: Configure the Pipeline to Automatically Trigger the Exploratory Testing Stage

The exploratory testing stage is automatically triggered when the commit stage finishes successfully. This means you must modify the commit stage build definition so that it knows to trigger the new stage.

1. From **Team Explorer – Builds**, open the **01 Commit Stage** build definition. Right-click on it and select **Edit Build Definition**.

2. Select the **Process** tab. Go to **section 5. Release pipeline** of the Build process parameters. Select **Next stages in pipeline**. Click the ellipsis **(…)**.



Team Foundation Build uses a build process template defined by a Windows Workflow (XAML) file. The behavior of this template can be customized by setting the build process parameters provided by the selected template.

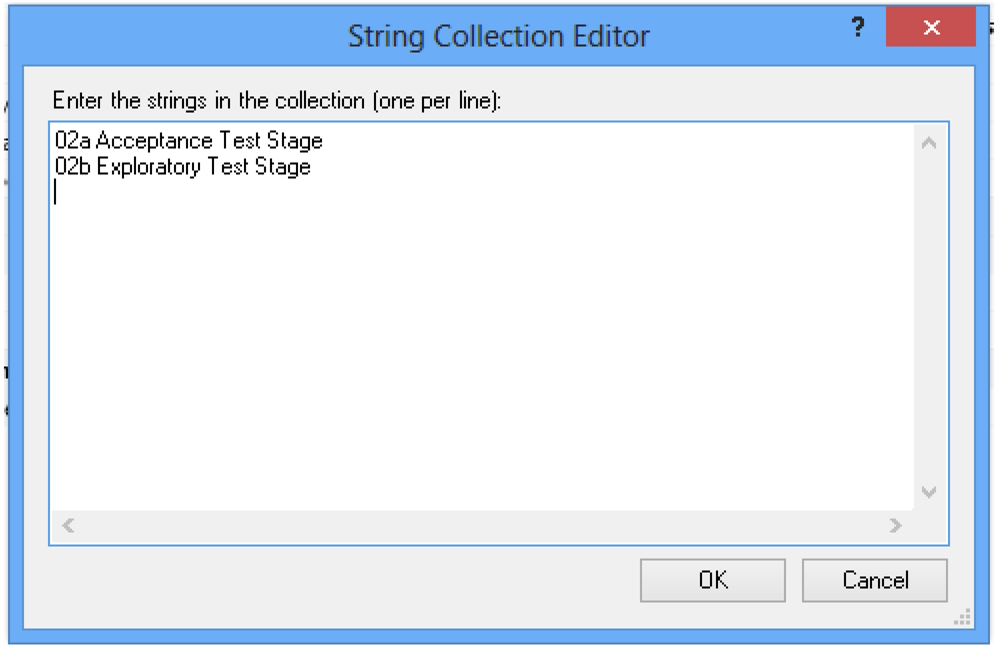
Build process template:

CDPipelineCommitStageProcessTemplate.xaml

Build process parameters:

1. Required	
Items to Build	Build 1 project(s) for 2 platform(s) and configuration(s)
2. Basic	
Automated Tests	Run tests in test sources matching **\*test*.dll, Test Case Filter
Clean Workspace	All
Logging Verbosity	Normal
Perform Code Analysis	Always
Source And Symbol Server Settings	Index Sources
3. Advanced	
4. Versioning	
Major Version	0
Minor Version	0
5. Release pipeline	
Next stages in pipeline	String[] Array ...

3. The **String Collection Editor** opens. Change the name of the acceptance test stage to **02a Acceptance Test Stage**. Add a new line with the name **02b Exploratory Test Stage**. Click **OK**. Save the build definition.


String Collection Editor

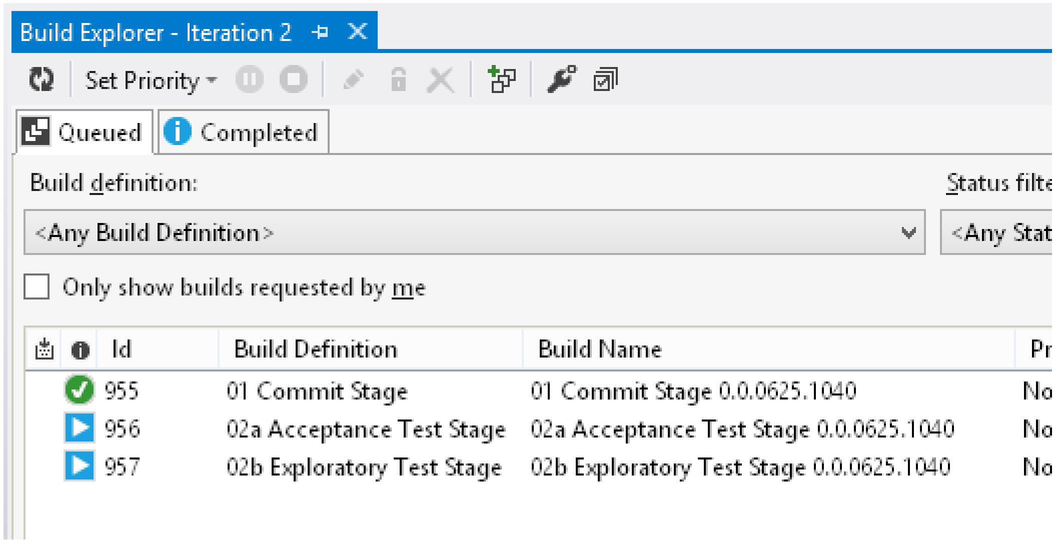
Enter the strings in the collection (one per line):

```
02a Acceptance Test Stage
02b Exploratory Test Stage
```

Task 3: Run and Monitor the Exploratory Test Stage

In this task you trigger a new pipeline instance and monitor it to see how it works with the new stage.

1. Trigger an instance of the pipeline by queuing the **01 Commit Stage** build definition. Make sure that Build Explorer is open so that you can monitor the progress of the pipeline instance.
2. After the commit stage finishes successfully, both the **02a Acceptance Test Stage** and the **02b Exploratory Test Stage** should be triggered.


Build Explorer - Iteration 2

Build definition: <Any Build Definition>

Status filte: <Any Stat

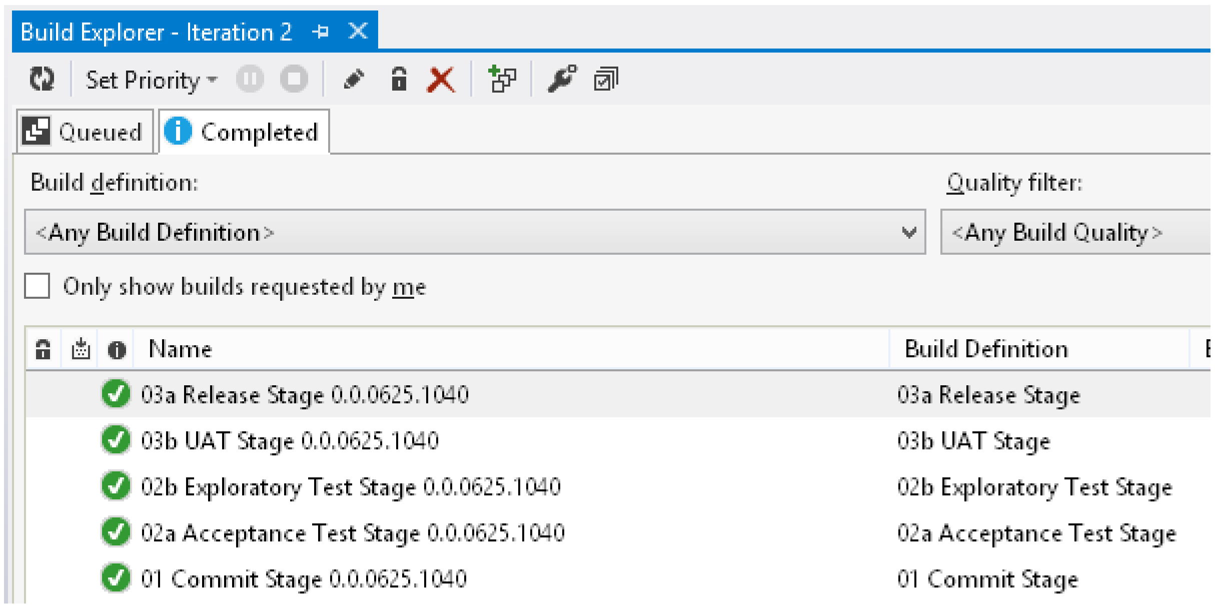
Only show builds requested by me

Id	Build Definition	Build Name	Pr
✓ 955	01 Commit Stage	01 Commit Stage 0.0.0625.1040	No
▶ 956	02a Acceptance Test Stage	02a Acceptance Test Stage 0.0.0625.1040	No
▶ 957	02b Exploratory Test Stage	02b Exploratory Test Stage 0.0.0625.1040	No

Using the Pipeline Test Stages

Once the exploratory test stage is triggered, you can immediately perform tests on any changes to the application because it is already deployed to the target environment and is ready to be used. You can also run the UAT stage so that users can begin to perform user acceptance tests. When all the tests are done, you can, if you want, manually trigger the release stage so that the fully tested application is available to end users.

The following screenshot shows a pipeline instance that has successfully completed all its stages.



Name	Build Definition
✓ 03a Release Stage 0.0.0625.1040	03a Release Stage
✓ 03b UAT Stage 0.0.0625.1040	03b UAT Stage
✓ 02b Exploratory Test Stage 0.0.0625.1040	02b Exploratory Test Stage
✓ 02a Acceptance Test Stage 0.0.0625.1040	02a Acceptance Test Stage
✓ 01 Commit Stage 0.0.0625.1040	01 Commit Stage

Summary

In this HOL you added an exploratory test stage to the continuous delivery pipeline that you implemented in the previous labs. This stage is automatically triggered by the commit stage. The new stage has one automated step, which is to deploy the application to the target environment. Tests that validate the application are performed manually.

To create the new stage, you adapted the existing acceptance test stage by modifying its build definition. You also modified the commit stage build definition to include the exploratory test stage in the list of automatically triggered stages.

Finally, you triggered an instance of the new pipeline and used Build Explorer to monitor its progress.

Copyright