DevOps with TFS

Lesson 03 : Build and Release Management

Capgemini

Table of Contents

Configuration Manager

- Configuration Manager is used to create, select, modify, or delete a configuration
- Different configurations of solution and project properties can be stored to use in different kinds of builds
- By default, Debug and Release configurations are included in projects that are created by using Visual Studio templates
- A Debug configuration supports the debugging of an app
- A Release configuration builds a version of the app that can be deployed

Solution Configuration

- A solution configuration specifies how projects in the solution are to be built and deployed.
- To modify a solution configuration or define a new one, in the **Configuration Manager**, under **Active solution configuration**, choose **Edit** or **New**.
- Each entry in the **Project contexts** box in a solution configuration represents a project in the solution.
- For every combination of **Active solution configuration** and **Active solution platform**, one can set how each project is used.
- The active solution configuration also provides context to the IDE.
  - For example, if you're working on a project and the configuration specifies that it will be built for a mobile device, the **Toolbox** displays only items that can be used in a mobile device project.

When you define a new solution configuration and select the **Create new project configurations** check box, Visual Studio automatically assigns the new configuration to all of the projects. Likewise, when you define a new solution platform and select the **Create new project platforms** check box, Visual Studio automatically assigns the new platform to all of the projects. Also, if you add a project that targets a new platform, Visual Studio adds that platform to the list of solution platforms and assigns it to all of the projects.
You can still modify the settings for each project.

Project Configurations

- The configuration and platform that a project targets are used together to specify the properties to use when it's built.
- A project can have a different set of property definitions for each combination of configuration and platform.
- Project configurations can differ considerably.
  - For example, the properties of one configuration might specify that its output file be optimized to occupy the minimum space, while another configuration might specify that its executable runs at the maximum speed.
- Project configurations are stored by solution—not by user—so that they can be shared by a team.
- The projects that are specified in the active solution configuration will be built.

To modify the properties of a project, you can use its Property Pages. (In **Solution Explorer**, open the shortcut menu for the project and then choose **Properties**.)
For each project configuration, you can define configuration-dependent properties as needed. For example, for a particular build, you can set which project items will be included, and what output files will be created, where they will be put, and how they will be optimized.
Although project dependencies are configuration-independent, only the projects that are specified in the active solution configuration will be built.

Authoring Build Definition

TFBuild – Team Foundation Build

- Extensible task-based execution system with a rich web interface
- Allows authoring, queueing and monitoring builds.
- Fully cross platform with underlying build agents.
- Integration with centralized version control system(TFVC) and distributed version control(Git Hub)
- Supports Work Item Integration, publishing Test execution result into TFS.
- All build definitions can easily be created in the web portal.
- The Web interface is accessible from any device and any platform

An agent is installable software that runs one build or deployment job at a time.
TFBuild is a task orchestrator that allows us to run any build engine, such as Ant,Maven,MSBuild,Visual Studio, Xamarin and so on.
The build agents are xCopyable and do not require any installation.

Hosted agents

If you're using Team Services, you've got the option to build and deploy using a **hosted agent**. When you use a hosted agent, Microsoft takes care of the maintenance and upgrades. So for many teams this is the simplest way to build and deploy. Hosted agents are available only in Team Services, not in Team Foundation Server (TFS).

Private agents

An agent that you set up and manage on your own to run build and deployment jobs is a **private agent**. You can use private agents in Team Services or Team Foundation Server (TFS). Private agents give you more control to install dependent software needed for your builds and deployments.
The agents are auto-updating in nature.

Authoring Build Definition

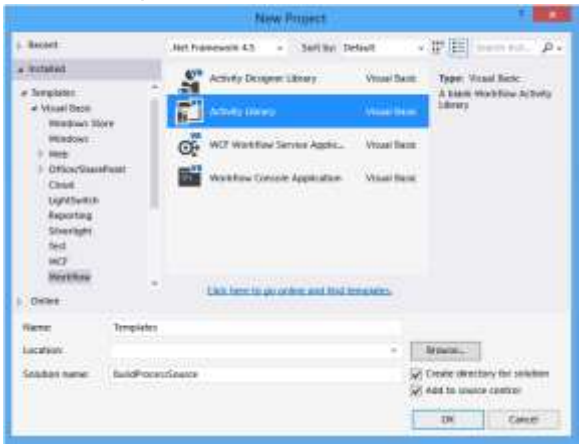
- Build Definition is a collection of Tasks
- Build Definition can be composed by dragging and dropping tasks.
  - A task is simply a build step
  - Each task supports Enabled, Continue on Error and Always Run flags.
  - By building a task, you're creating a reusable build component that can be shared with anyone using your technology.

Customizing the Build Process

- The Team Foundation Build activities are the fundamental components of the build process in TFBuild system.
- If you need your build process to do more than what the default template can do, you can customize the build process template to follow your own Windows Workflow Foundation (WWF) instructions. Your instructions can run .NET Framework code that is implemented in CodeActivity objects.
- You can run activities that are built into Team Foundation Build (TFBuild),
- The best way to create a custom build process template is to base it on the Default Template (**DefaultTemplate.xaml**).

Customizing the Build Process

Custom build process solution

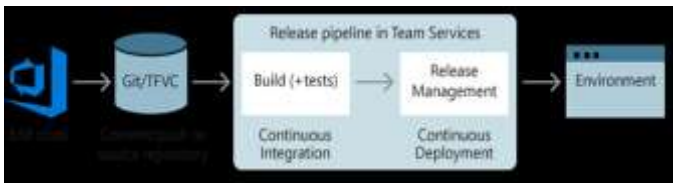


Customizing the Build Process

- Steps for creating and applying Custom Build Process
  - Start a custom build process solution and create a template
  - Add the Activities with required property settings and Save the Template
  - Use the Custom Template in the Build Definition
  - Queue the build
  - View the results

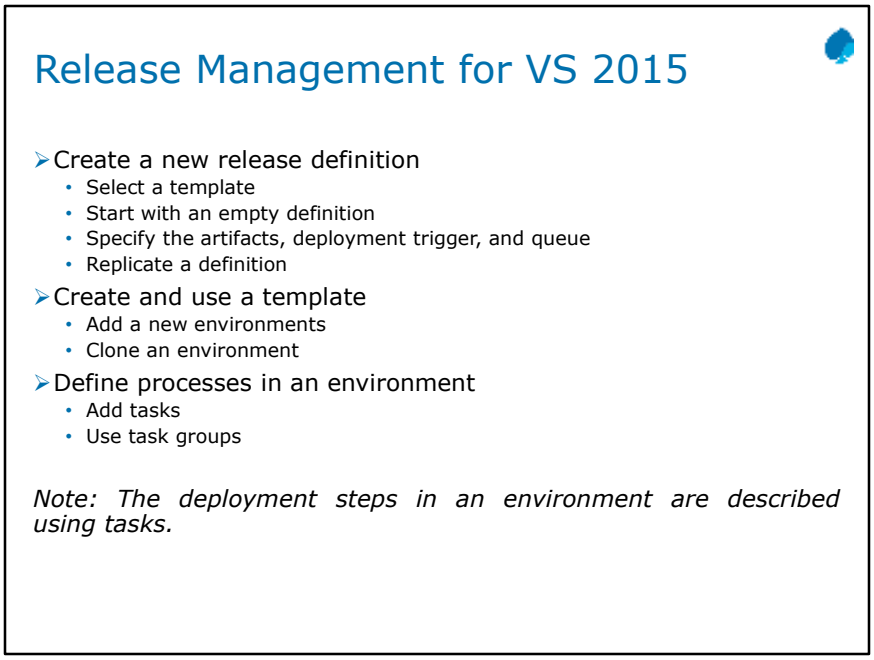
Release management – An Overview

- Release management is the process of managing, planning, scheduling and controlling a software build through different stages and environments; including testing and deploying software releases. - *Definition from Wikipedia*
- Release Management continuously deploys your app to a specific environment for each separate stage: development, test, staging, and production.
- Start your release process manually or from a build. Then track your releases as they move through your release path.



Release Management for VS 2015

- Release Management features have been integrated into the Build & Release hub in Team Foundation Server (TFS) and VSTS
- Support a rapid release cadence and manage simultaneous releases.
- Set up release paths that represent your stages from development to production.
- Run actions to deploy your app to an environment for that stage.
- Add approvers to sign off that the app has successfully passed each stage.



Create a release definition
Add one or more environments to release definition.
Add tasks to each environment.
Add approvals or make them automated, for each environment.
Save release definition
Start a release
Get the ID of the release definition that you want to use.
Create a release.
Get the ID of the release from the response so you can use it later.
If required, abandon a release.
Get a release details
Get a list of releases and find the ID of the release you're interested in.
Get the details about the release.
Get the approvals required for the release.
Accept/Reject approvals
For each environment where the application is being deployed, you can have pre-deployment or post-deployment approvals. If you are one of the approvers, you will get an approval request which you can accept or reject based on some criteria.

Release Management for VS 2015

- Release Management shows comprehensive information about the releases you have initiated, and the results of deployment, which includes
  - A list and an overview of all releases
  - A summary of the details for each release
  - Test results and
  - Release logs.
- From the Release Management UI you can also
  - restore deleted releases
  - respond to approval requests
  - redeploy a release
  - send email notifications and
  - view release history..

Summary

- Understanding the Configuration of Build Environment
- Types of Build Configuration
  - Solution
  - Project
- Authoring Build Definitions with TFBuild
- Creating Custom Build Definition
- Understanding the Release Management
- Creating and Applying Release Definitions