



Automating Builds with Team Foundation Server 2008

Martin Woodward
Senior Software Engineer
Teamprise

imtc2008

Why are you here?

- You've used Team Build 2005 and want to know what's new in 2008
- You are unsure about build automation and wonder if Team Build can help
- You're a believer in build automation and agility and really wonder, does Microsoft get it

Agenda

1. Creating and Using a Team Build
2. Customizing the Build Process
3. Using the Build API

Completed version of demo available at:

<http://code.msdn.microsoft.com/buildwallboard>

imtc2008

Survey

- Who is using TFS
- Who is using Team Foundation Build
- Who is using CC.NET etc?

Build Automation is not ...



F5

imtc2008

Build Automation is ...

- Collecting
- Assembling
- Validating
- Auditing

Build automation helps you answer the tough questions

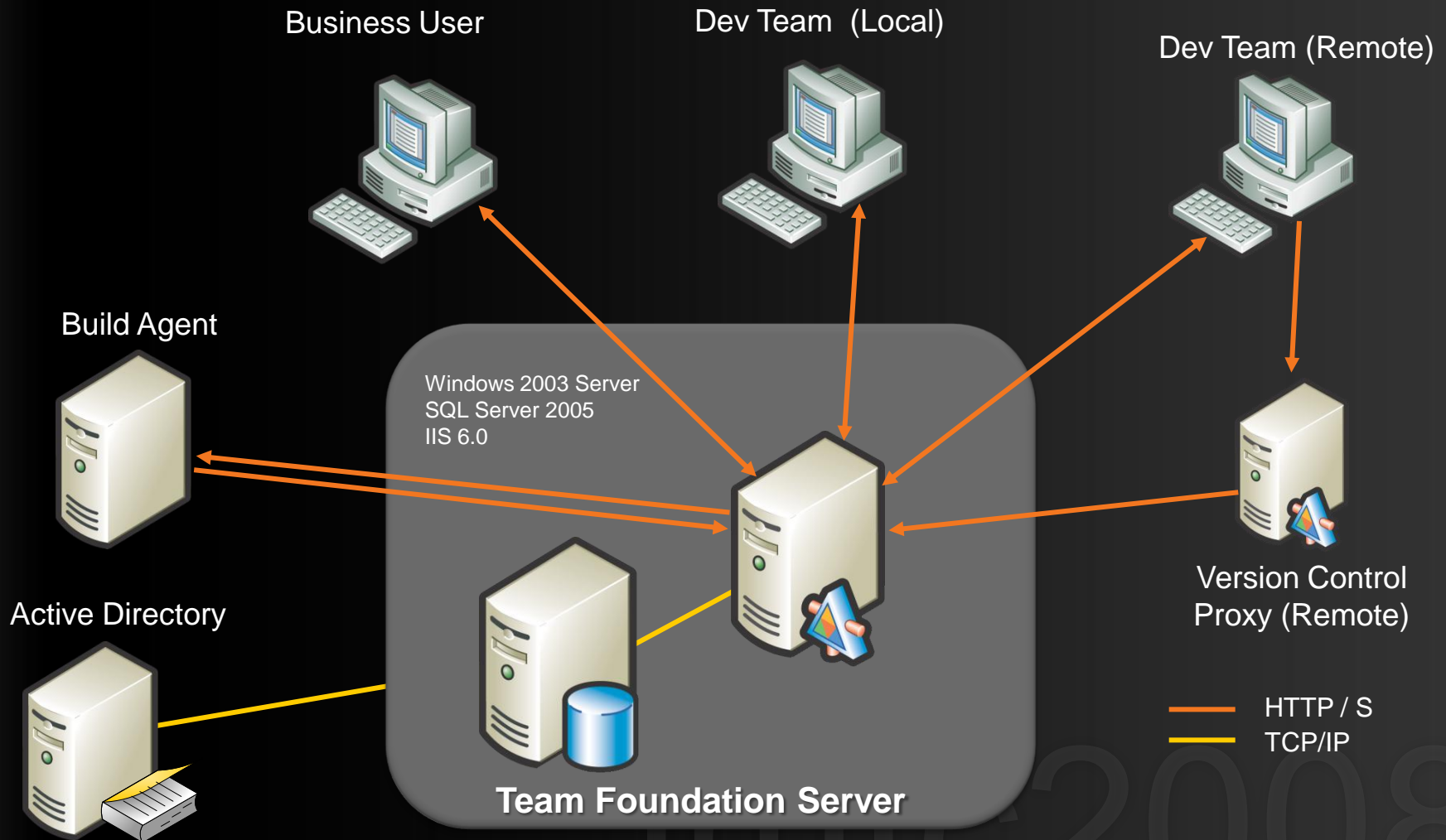
- Do our bits create a product?
- Are we on schedule?
- Is our product high quality?

Build automation's value is in making it trivial to take the pulse of your team

Team Build 2008

- Visual Studio 2008 Team Foundation Server Build is the official name of the feature
- Core feature of TFS—you don't buy it on its own
- Industrial strength build automation in Team System
- Provides the “F5” experience for your team

Team Foundation Server





Demo

Using Build in Visual Studio Team Foundation Server 2008

imtc2008

Key Features in Team Build 2005

- Reports for status and other quality metrics
 - Unit test results
 - Static Analysis results
 - Associated work items and change sets
 - Code coverage
- Warehouse support for historical trends
- Multiple build types
- Multiple remote build machines
- Ability to kick off a build on demand via UI or command line
- Build notifications

New Team Build 2008 Features

- Continuous Integration
- Build Queuing
- Scheduled Builds
- Build Agent Management
- Build Definition Editing GUI
- Better Build Management
- Managed object model
- Improved extensibility

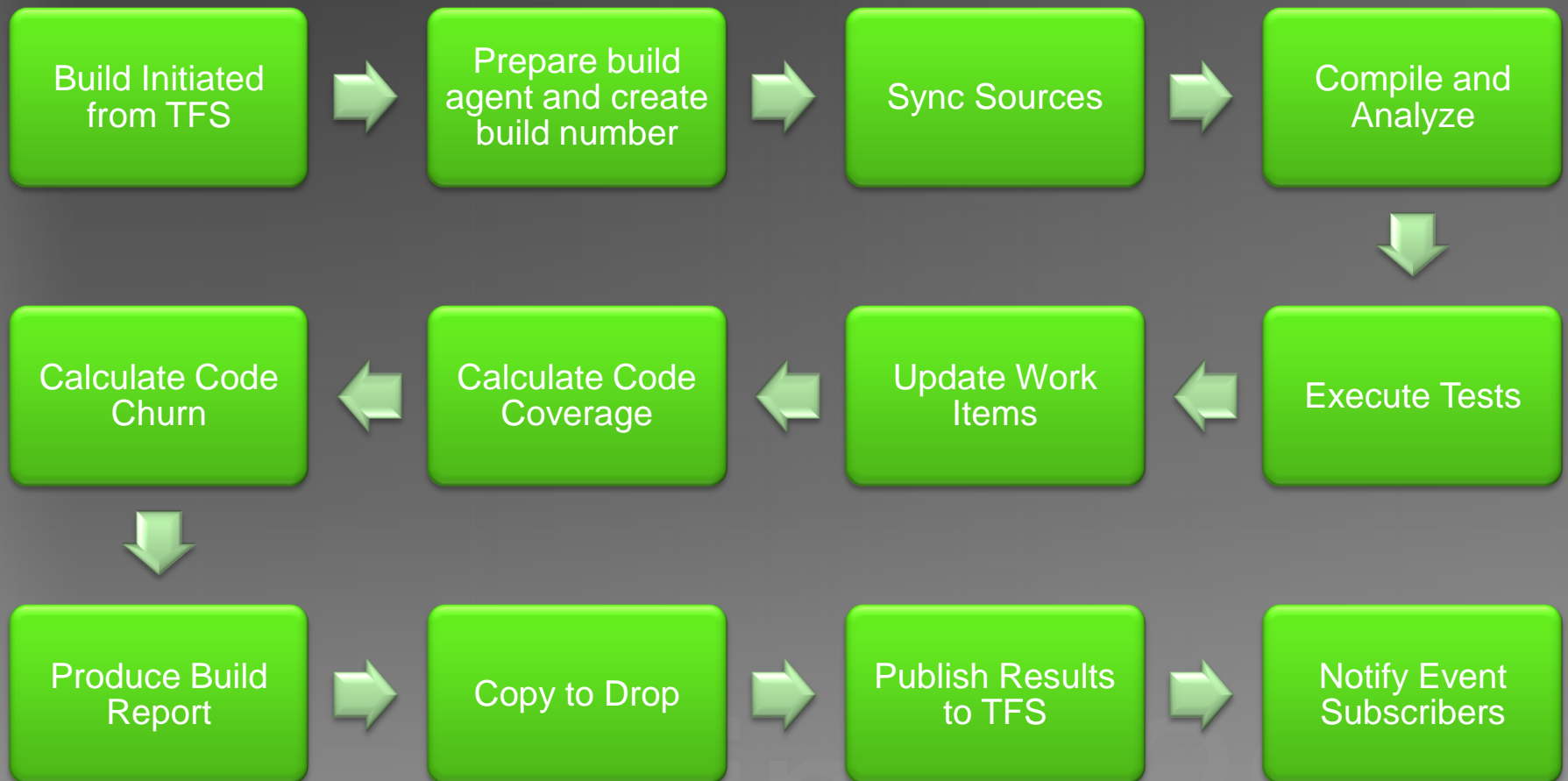
Team Build 2008 / 2005 Compatibility

Feature	TFS2005	TFS2008
Create Build	5	8
Start / Queue Build	5	8
View Build Definitions	5	8
View Completed Builds	5	8
View Queued Builds		8
View Build Reports	5	8
View Build Log	5	8
Edit Build Quality	5	8
Open Drop Folder	5	8
Stop/Cancel Build		8
Pass command line arg		8

Feature	TFS2005	TFS2008
Delete Build Definition		8
Edit Retention Policy		8
Edit Build Triggers		8
Assign Build Agent (UI)	5	8
Delete Build in UI		8
Build .NET 2.0 project	5	8
Build .NET 3.5 project		8
Postpone Build		8
Manage Build Qualities	5	8
Manage Build Agents		8
Edit workspace map	5	8

Build Execution

BuildDefinition \ TFSBuild.proj



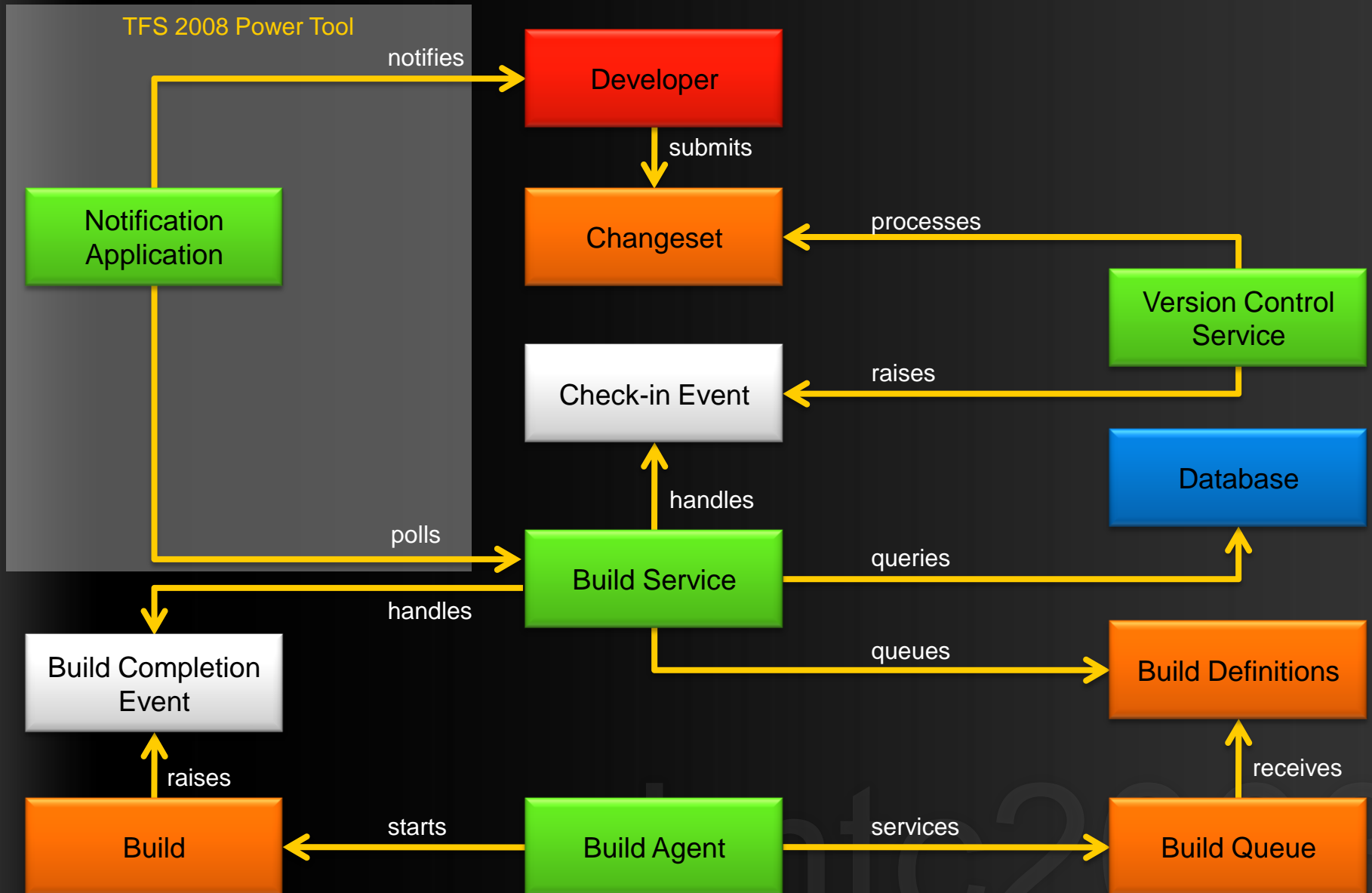
Continuous Integration

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day.

Martin Fowler

imtc2008

How CI Works in 2008



Run Flexibility

- Ability to kick off a build on demand via Visual Studio UI or from the command line
- New Continuous Integration support and GUI management of builds
- Support for scheduled builds
- Managed API also exposed for rich control

Multiple Remote Build Machines

- Team Build is designed to support multiple build machines (build agents)
- Chosen via UI or command-line when build is requested
- TFS 2008 supports easily adding new build agents without creating new builds
- Can even have multiple build agents running on one machine

Build Management

- Manual process in TFS 2005
 - Command-line deletion of builds
 - Manual management of drops
- Team Build 2008 provides GUI features to manage builds and artifacts
- Team Build 2008 supports retention policy for builds and automatic clean up

Build Notifications

- Using the project alerts feature, anyone can subscribe to build notification events
- Two events supported out of the box
 - E-mail notification generated
 - Build status changes
 - Build completes
- Build Notification Power Tool



Part 2

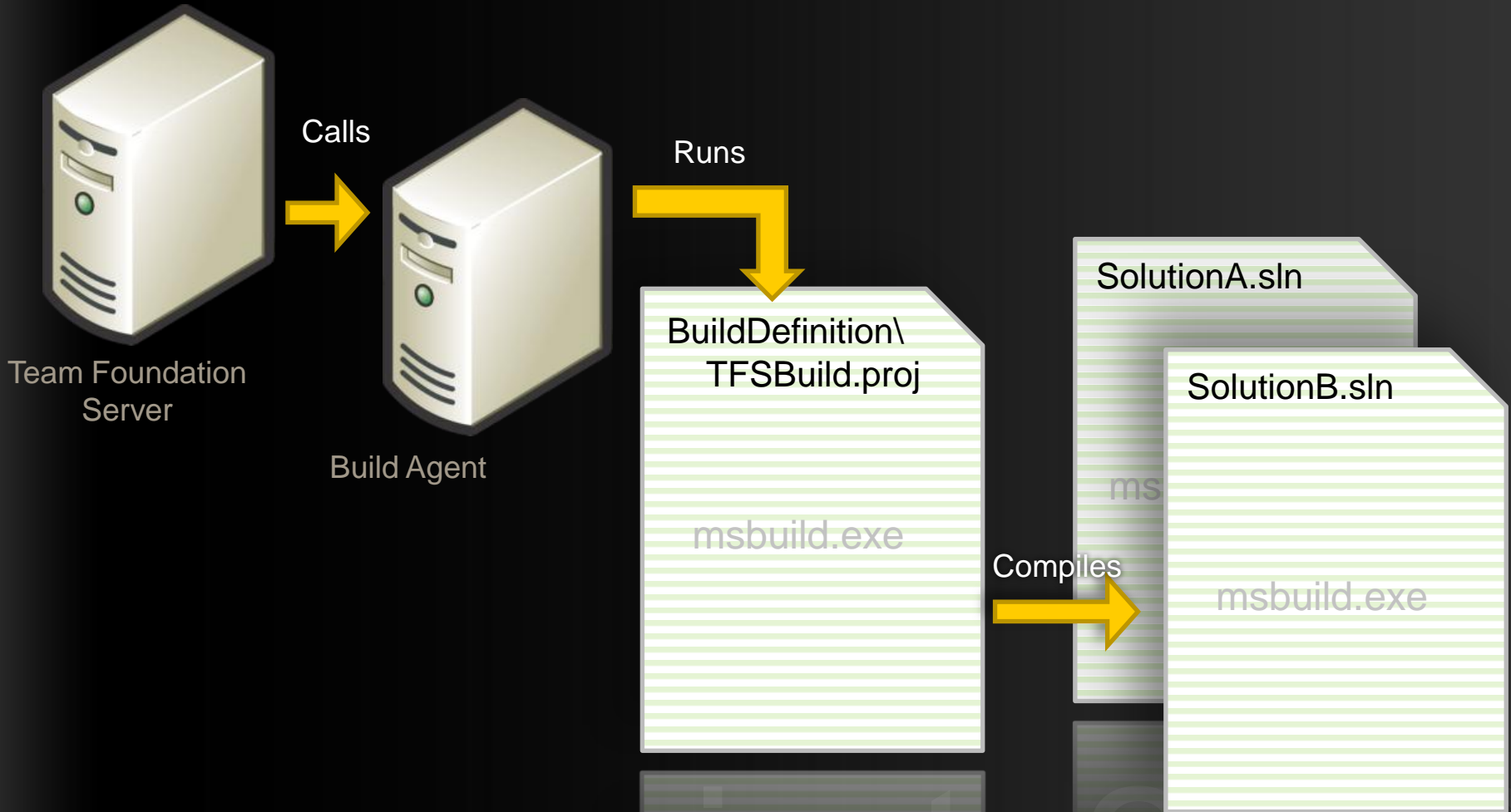
Customizing the Build Process

imtc2008

Team Build Architecture

- Team Build uses MSBuild engine
- Consists of
 - MSBuild tasks for Team System components (testing, code churn, update work items etc.)
 - MSBuild project file (.proj), created via GUI, that defines build options.

Team Build Architecture



Customizing the Build

- Core build defined in TfsBuild.proj
- Check-out for edit and modify
- Targets group tasks together in a particular order
 - Allow sections of the build process to be called from the command line
 - Targets contain tasks
- Tasks
 - Tasks are defined operations that live in a managed assembly
 - They provide the code that runs during the build process
 - You can write your own
 - Implement ITask
 - Derive from helper class Task

Customizing the Build

- Properties pre-defined to control build

IncrementalGet	StopOnFirstFailure
IncrementalBuild	CleanCompilationOutputOnly
SkipWorkItemCreation	UpdateAssociatedWorkItemsOnBuildBreak
SkipLabel	SkipGetChangesetsUpdateWorkItems

- Targets in Build designed to be overridden

BeforeOnBuildBreak	BuildNumberOverrideTarget
BeforeCompile	BeforeEndToEndIteration
BeforeTest	AfterEndToEndIteration
PackageBinaries	Before/After everything!

Desktop Builds

- Build scripts work on desktops also
- Can install build server locally for UI support and publication
- Can use command-line for private, personal builds

Only 3 Things You Need To Know

- MSDN Help—Understanding Team Foundation Build Configuration Files

- [http://msdn2.microsoft.com/en-us/library/ms400710\(VS.90\).aspx](http://msdn2.microsoft.com/en-us/library/ms400710(VS.90).aspx)

- Microsoft.TeamFoundation.Build.targets file

- %ProgramFiles%\MSBuild\Microsoft\VisualStudio\TeamBuild\
Microsoft.TeamFoundation.Build.targets

!! NEVER EDIT THIS FILE !!

- Many custom tasks already published—search before you write your own

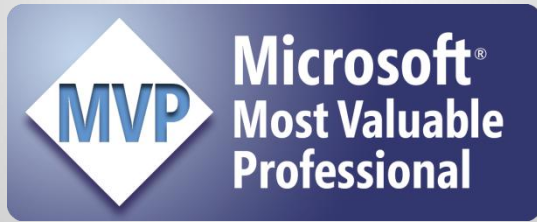
imtc2008



Demo

Customizing the Build Process

imtc2008



Part 3

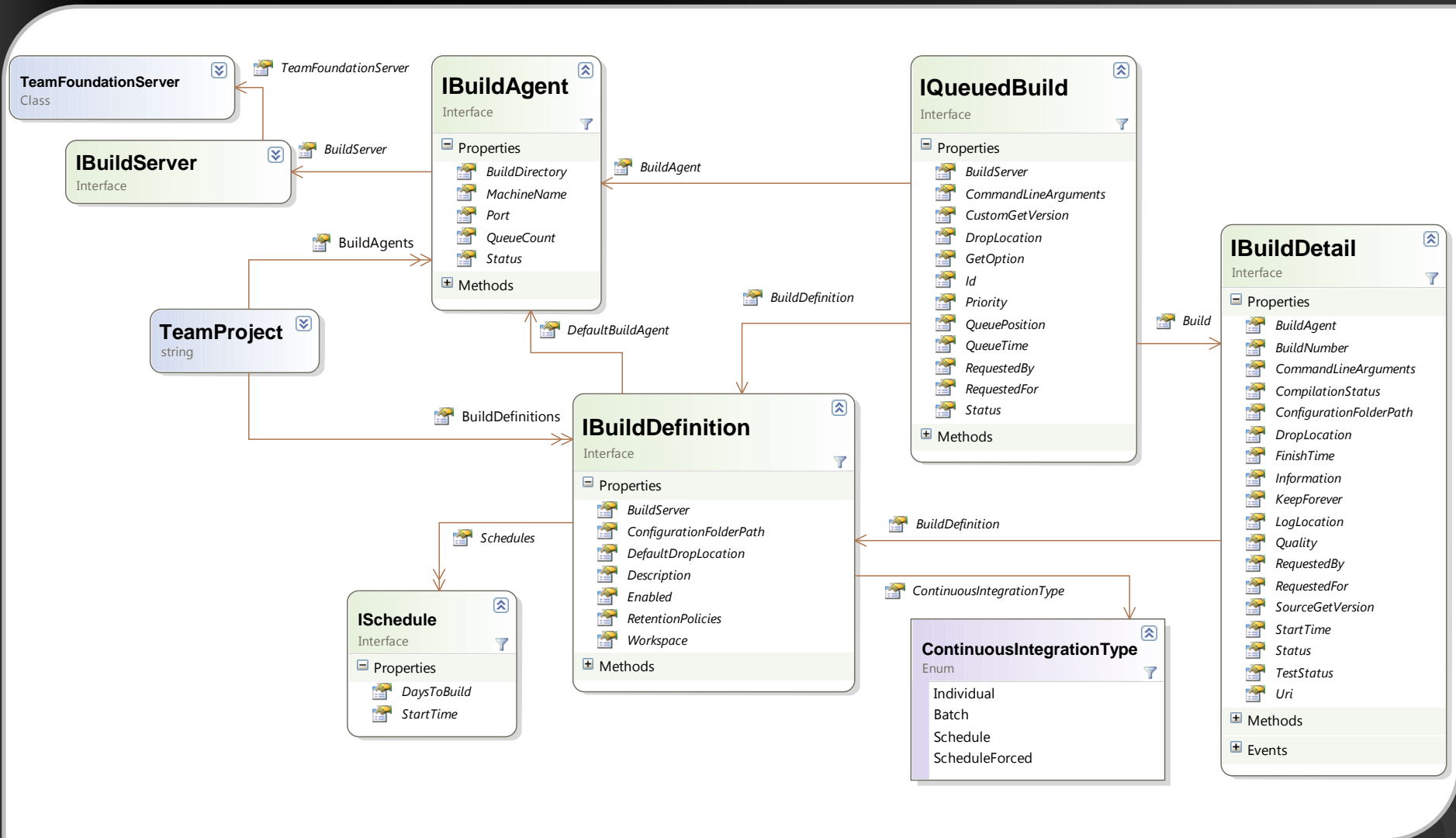
Using the Build API

imtc2008

Team Build Managed API

- Access to same API used by Team Build
- Allows you to automate everything done in UI & more
- Can build integration with existing build systems (Nant, Ant, make etc) into Team Build

Team Build Managed Code API



Team Build API Example

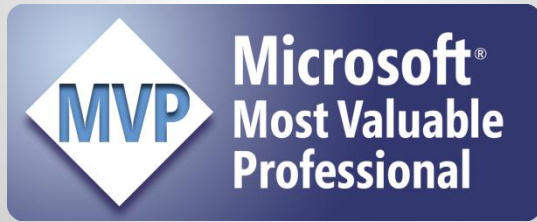
C#

```
TeamFoundationServer tfs =  
    TeamFoundationServerFactory.GetServer("http://tfsserver:8080");  
  
IBuildServer buildServer =  
    (IBuildServer) tfs.GetService(typeof (IBuildServer));  
  
IBuildDefinition buildDef =  
    buildServer.GetBuildDefinition("TeamProject", "Build Name");  
  
buildServer.QueueBuild(buildDef);
```

VB.NET

```
Dim tfs As TeamFoundationServer =  
    TeamFoundationServerFactory.GetServer("http://tfsserver:8080")  
  
Dim buildServer As IBuildServer =  
    DirectCast(tfs.GetService(GetType (IBuildServer)), IBuildServer)  
  
Dim buildDef As IBuildDefinition =  
    buildServer.GetBuildDefinition("TeamProject", "Build Name")  
  
buildServer.QueueBuild(buildDef)
```

imtc2008



Demo

Using the Managed Code API for Team Build 2008

imtc2008

Summary

- Team Build allows for consistent and repeatable builds
- MS Build engine support provides for limitless extensibility
- Visual Studio 2008 Team Foundation Server version of Team Build provides dramatic improvement



Q&A

Martin Woodward

martin@teamprise.com

<http://www.woodwardweb.com>

<http://code.msdn.microsoft.com/buildwallboard>

imtc2008

Next Sessions

Track	Description
1	Securing Communications using WCF (Chris Seary)
2	Improving Team Development with Microsoft Visual Studio Team System 2008 (Doug Seven)
3	Windows Mobile & Systems Centre Mobile Device Manager (Mark Mulvany)
4	LINQ to SQL - An Introduction (Ronan Geraghty)

Martin Woodward

martin@teamprise.com

<http://www.woodwardweb.com>

<http://code.msdn.microsoft.com/buildwallboard>