Build Automation

Epic's Build Tools & Infrastructure

Gerke Max Preussner

max.preussner@epicgames.com

Sorry, we ran out of cats for this presentation

Build Tools

UHT

UBT

UAT

Unreal Header Tool (UHT)

- Written in C++
- Parses all C++ headers containing UClasses
- Generates glue code for all Unreal classes & functions
- Generated files stored in Intermediates directory

Build Tools

UHT

UBT

UAT

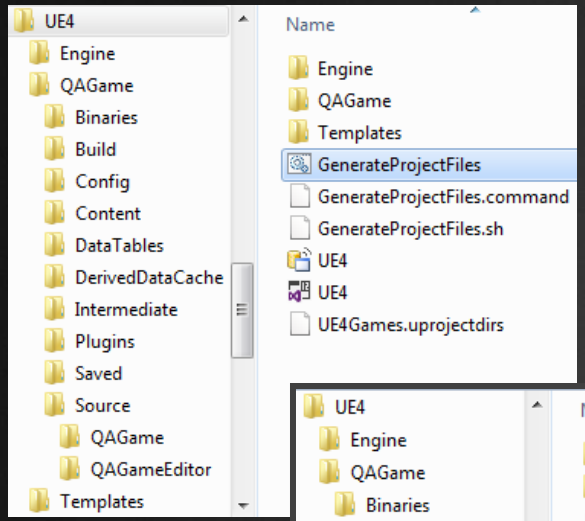
Unreal Build Tool (UBT)

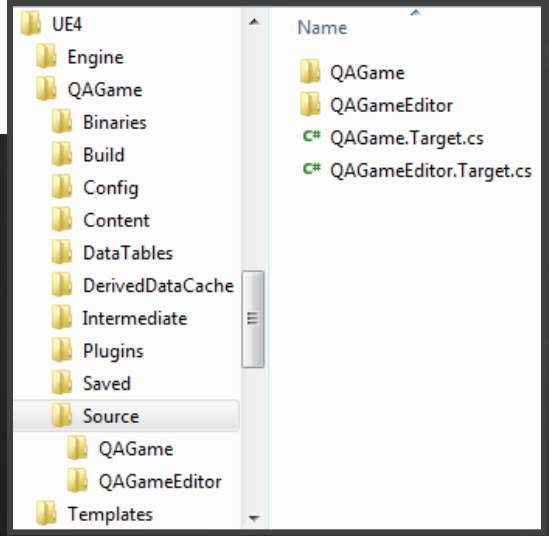
- Written in C# (may convert to C++ in the future)
- Scans solution directory for modules and plug-ins
- Determines all modules that need to be rebuilt
- Invokes UHT to parse C++ headers
- Creates compiler & linker options from .Build.cs & .Target.cs
- Executes platform specific compilers (VisualStudio, LLVM)

Other UBT Features

- Project file generation (GenerateProjectFiles.bat)
- Remote Compilation (iOS, MacOS)

Build Rules

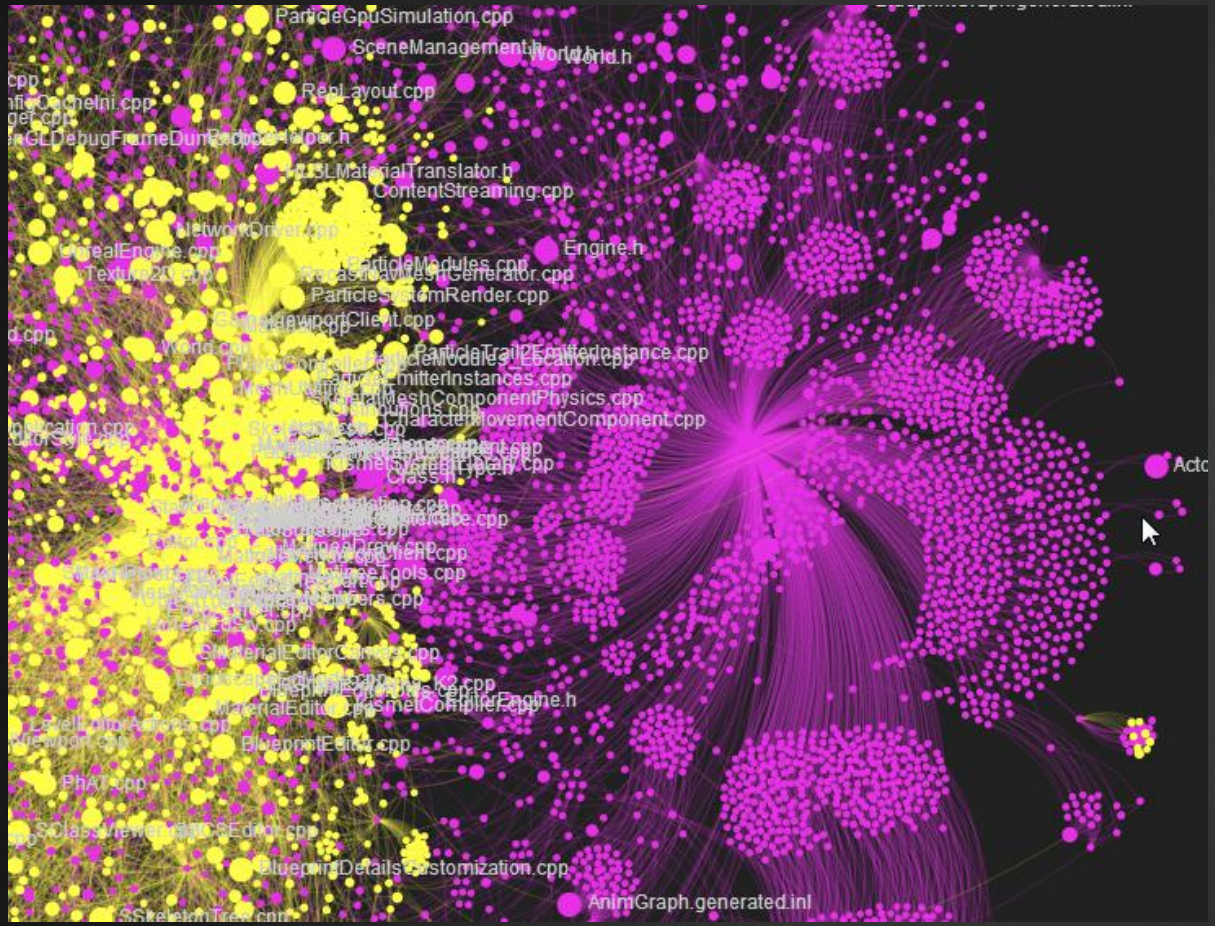





```
// Copyright 1998-2014 Epic Games, Inc. All Rights Reserved.

using UnrealBuildTool;

public class QAGame : ModuleRules
{
    public QAGame(TargetInfo Target)
    {
        PublicDependencyModuleNames.AddRange(
            new string[] {
                "Core",
                "CoreUObject",
                "Engine",
                "HTTP",
                "OnlineSubsystem",
                "OnlineSubsystemUtils"
                "GameplayAbilities",
                "AIModule",
            }
        );

        PrivateDependencyModuleNames.A
            new string[] {
                "RenderCore",
                "Slate",
                "SlateCore",
            }
        );

        if ((Target.Platform == Unreal
```

```
// Copyright 1998-2014 Epic Games, Inc. All Rights Reserved.

using UnrealBuildTool;
using System.Collections.Generic;

public class QAGameTarget : TargetRules
{
    public QAGameTarget(TargetInfo Target)
    {
        Type = TargetType.Game;
    }

    //
    // TargetRules interface.
    //

    public override void SetupBinaries(
        TargetInfo Target,
        ref List<UEBuildBinaryConfiguration> OutBuildBinaryC
        ref List<string> OutExtraModuleNames
        )
    {
        OutExtraModuleNames.Add("QAGame");

        if (UEBuildConfiguration.bBuildEditor)
        {
            OutExtraModuleNames.Add("QAGameEditor");
        }
    }
    public override List<UnrealTargetPlatform> GUBP_GetPlatf
```

Dependency Graph



Build Tools

UHT

UBT

UAT

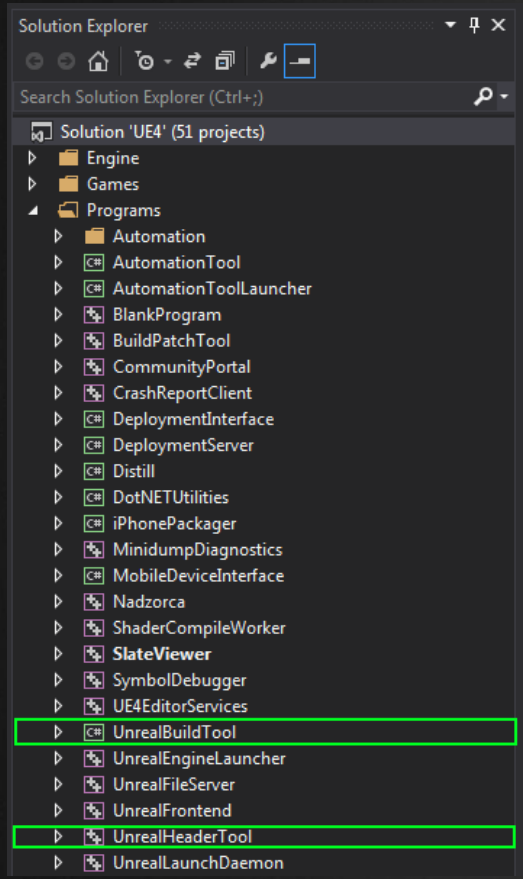
Unreal Automation Tool (UAT)

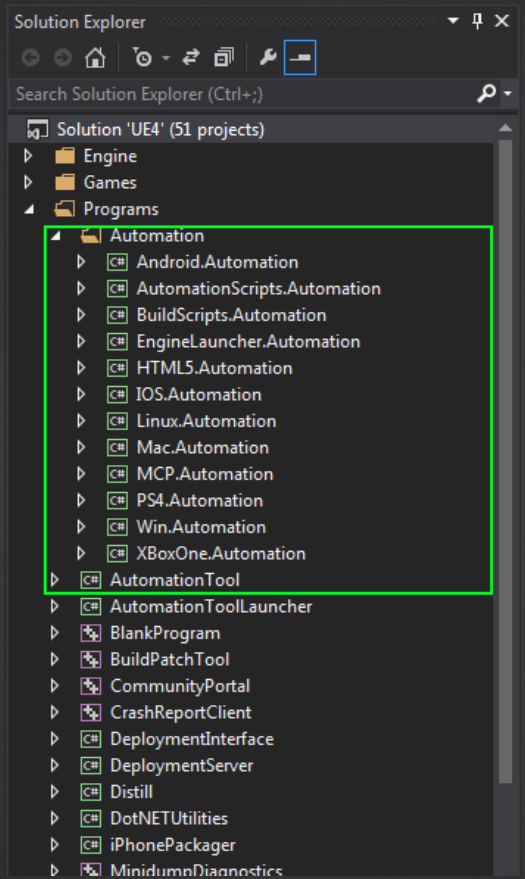
- Written in C# (may convert to C++ in the future)
- Automates repetitive tasks through Automation Scripts
- Build, cook, package, deploy and launch projects
- Invokes UBT for compilation

Other UAT Scripts

- Analyze and fix up game content files
- Code surgery when updating to new Engine versions
- Distributed compilation (XGE) & build system integration
- Generate code documentation
- Automated Testing of code and content
- And many others – you can add your own scripts!

UBT, UHT, UAT Projects





Build Automation

SCC

CIS

Promotion

Testing

Source Code Control (SCC)

- We use Perforce, but you don't have to
- Not just code, but also content and everything else
- Also contains compiled binaries from build system (!)
- Used to version many other things (Markting, QA, etc.)

GitHub Integration

- Check-ins are pushed to GitHub Master in near real-time
- Script for converting GitHub pull requests to P4 CLs

Build Automation

SCC

CIS

Promotion

Testing

Continuous Integration System (CIS)

- Verifies all check-ins of code and content
- Grand Unified Build Process (GUBP)
- Thousands of build tasks a day

Backend Software

- Used custom build server for several years
- Experimented with Jenkins, but not scalable enough
- Now using Electric Commander
- Complex workflows with hundreds of jobs and sub-tasks

Build Hardware

- Virtualized farm of Windows and MacOS build servers

GUBP

electriccommander

Logged in as 'Max Preussner' | Logout | Help

Home | CIS | Promotions | Rocket | Projects | Jobs | UnTab | Workflows | Cloud | Artifacts | Search | Administration

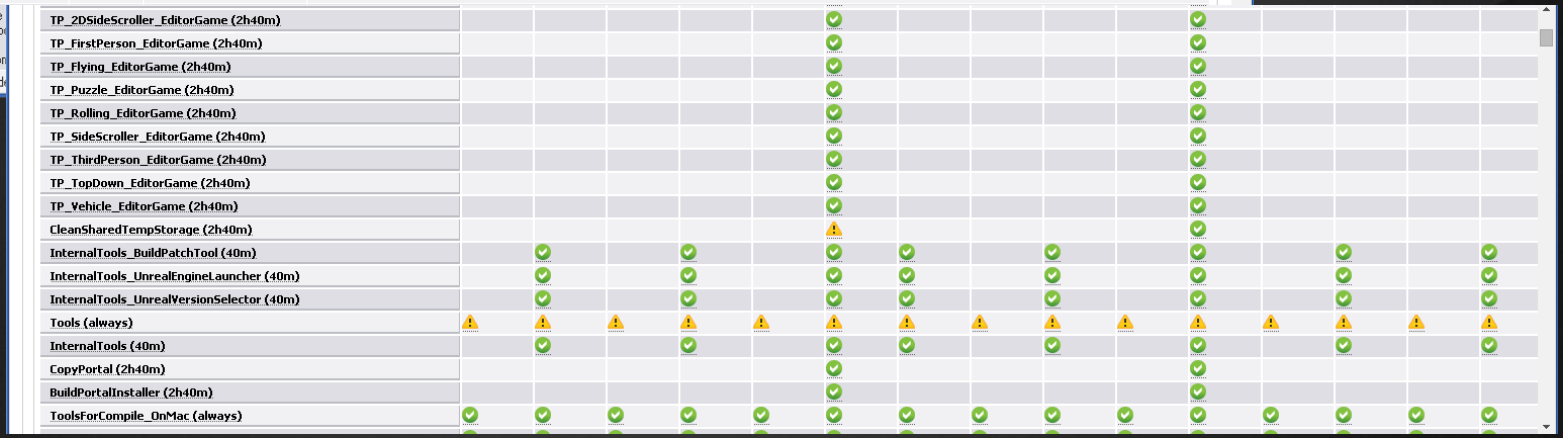
Project Details – GUBP

Edit | Access Control | 13/29

Procedures | Workflow Definitions | Jobs | Workflows | Schedules | Credentials | Properties | Reports

Create Procedure

Procedure	Description	Resource	Create Date	Actions
AgentSetup	Acquires a resource, bootstraps the agent, sync and sets up the branch info		2014-01-09 08:32:34 EST	Run | Edit | Copy | Delete
Editors	Compile the base editor, tools, game editors and platform editors		2013-12-02 15:57:52 EST	Run | Edit | Copy | Delete
Game Monolithic Platform	Compiles all game monolithics for a single platform		2014-01-10 13:40:50 EST	Run | Edit | Copy | Delete
Game Monolithic Platform Test	Tests a monolithic platform		2014-01-10 13:40:50 EST	Run | Edit | Copy | Delete
Game Platform Cook	Cooks everything for a single platform		2014-01-10 13:40:50 EST	Run | Edit | Copy | Delete
Game Promote	Just an aggregate maybe a label), no like: FortniteGame_Pron			
Game Uncooked Tests	Tests a game, node			



UNREAL ENGINE

Build Automation

SCC

CIS

Promotion

Testing

Build Promotion

- Selected successful CIS builds are tested by QA
- Builds that pass QA will be promoted to stable
- Selected stable builds become release candidates
- Approved stable builds are released to public

Build Automation

SCC

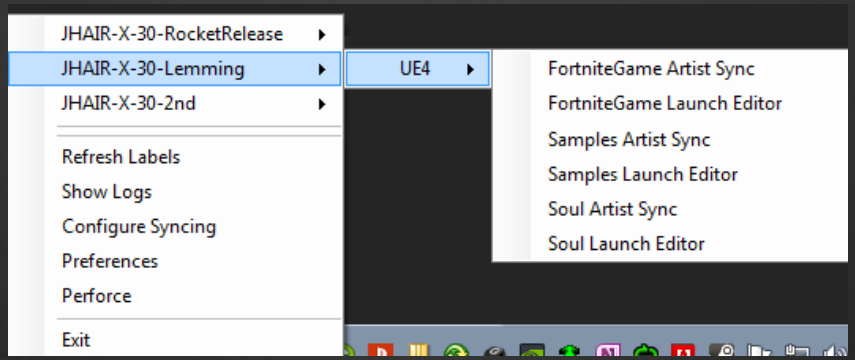
CIS

Promotion

Testing

UnrealSync

- Tool for artists to fetch the latest promoted build
- Aware of P4 branches and projects
- Notifies user when new promoted build is available



Build Automation

SCC

CIS

Promotion

Testing

Automated Testing

- Simple and complex tests
- Unit tests for C++, content verification
- Parallel testing on multiple platforms & devices
- Can also run from command line (as a Commandlet)

Simple Tests

- Single atomic test that can pass or fail
- Unit tests for C++, feature tests for content
- Examples: Play a map in PIE, verify text wrapping in Slate, etc.

Complex Tests

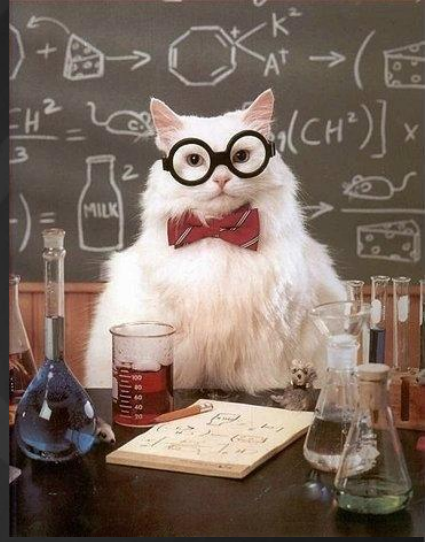
- Run same test code on a number of inputs
- Examples: Load all maps in the Editor, compile all Blueprints, etc.

Simple Automation Test

```
// Example for a simple automation test that runs in a game

IMPLEMENT_SIMPLE_AUTOMATION_TEST(FSetResTest, "Windows.SetResolution", ATF_Game)

bool FSetResTest::RunTest( const FString& Parameters )
{
    FString MapName = TEXT("AutomationTest");
    FEngineAutomationTestUtilities::LoadMap(MapName);

    int32 ResX = GSystemSettings.ResX;
    int32 ResY = GSystemSettings.ResY;
    FString RestoreResolutionString = FString::Printf(TEXT("setres %dx%d"), ResX, ResY);

    ADD_LATENT_AUTOMATION_COMMAND(FEngineWaitLatentCommand(2.0f));
    ADD_LATENT_AUTOMATION_COMMAND(FExecStringLatentCommand(TEXT("setres 640x480")));
    ADD_LATENT_AUTOMATION_COMMAND(FEngineWaitLatentCommand(2.0f));
    ADD_LATENT_AUTOMATION_COMMAND(FExecStringLatentCommand(RestoreResolutionString));

    return true;
}
```

Automation Test Demo

Questions?



Documentation, Tutorials and Help at:

- AnswerHub: http://answers.unrealengine.com
- Engine Documentation: http://docs.unrealengine.com
- Official Forums: http://forums.unrealengine.com
- Community Wiki: http://wiki.unrealengine.com
- YouTube Videos: http://www.youtube.com/user/UnrealDevelopmentKit
- Community IRC: #unrealengine on FreeNode

Unreal Engine 4 Roadmap

- lmgtfy.com/?q=Unreal+engine+Trello+