XCOM 2: SDK Quick Start

Contents

Introduction

The XCOM2 SDK for making mods consists of three main components:

1. Firaxis' Modbuddy, with XCOM2 extensions
2. XCOM2's 64 bit version of UnrealEd.
    a. This executable is also capable of running UE3 commandlets such as the script compiler or cooker.
3. XCOM2's shipping content
    a. Uncooked maps and packages
    b. Unreal script source code

This quick start guide will focus on getting Modbuddy up and running, and demonstrate how to use it to open and publish an example mod.

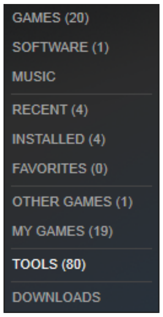
Getting Started

The first step creating an XCOM2 mod is to launch Modbuddy.

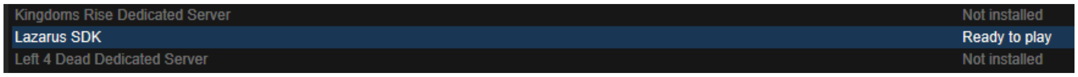
Install the XCOM2 SDK on Steam

The XCOM2 SDK is available on steam under Library->Tools.

Navigate to Tools:



Find the XCOM2 SDK under the name "XCOM 2 SDK" and select Install from the right click menu option.



Kingdoms Rise Dedicated Server	Not installed
Lazarus SDK	Ready to play
Left 4 Dead Dedicated Server	Not installed

Install the SDK redistributables

Modbuddy for XCOM2, similar to previous versions of Modbuddy released for Civilization, uses the visual studio isolated shell to provide an IDE for modding. You can find the redistributables in the game's installation directory under the "Binaries" folder. Example: "\SteamLibrary\steamapps\common\XCOM 2 SDK\Binaries\Redist". There are two redistributables provided: one for visual studio and one for UE3, please install both.

Launch Modbuddy for XCOM2

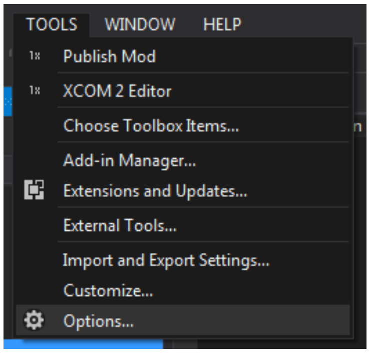
Choose "Play Game" on the XCOM 2 SDK in steam and this will launch Modbuddy for XCOM2. The IDE may take a while to load for the first time.

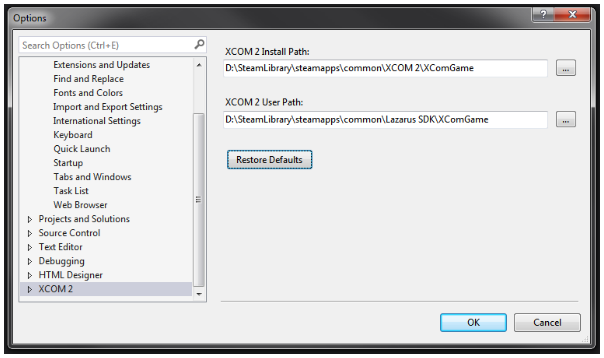
First Time Setup

After launching Modbuddy for the first time, users will want to set general settings on the IDE related to where XCOM2 is installed. Choose Tools->Options, and then scroll down to the bottom of the options menu as shown below. Verify that the paths are set correctly.

XCOM 2 Install Path – This path should point to the **GAME** installation, ending in XComGame.

XCOM 2 User Path – This path should point to the **SDK** installation, ending in XComGame.

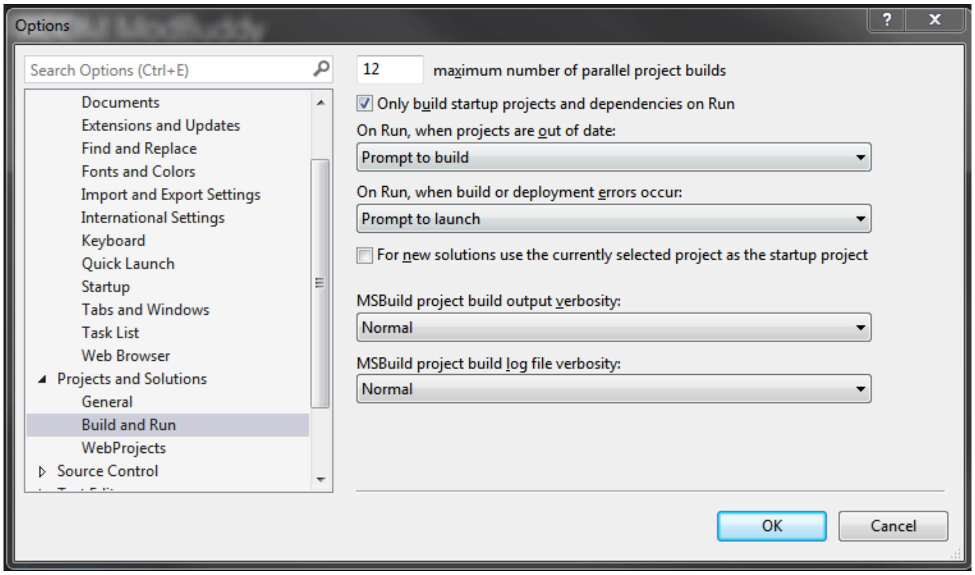




XCOM 2 Install Path:
D:\SteamLibrary\steamapps\common\XCOM 2\XComGame

XCOM 2 User Path:
D:\SteamLibrary\steamapps\common\Lazarus SDK\XComGame

By default, build information in the Visual Studio Isolated Shell is set to “minimal’. To get information from the build process / script compiler, you will want to change this setting. Choose Tools -> Options -> Projects and Solutions -> Build and Run. Change the two verbosity dropdowns at the bottom of the options panel to “Normal”, then hit OK.



12 maximum number of parallel project builds

☑ Only build startup projects and dependencies on Run

On Run, when projects are out of date:
Prompt to build

On Run, when build or deployment errors occur:
Prompt to launch

☐ For new solutions use the currently selected project as the startup project

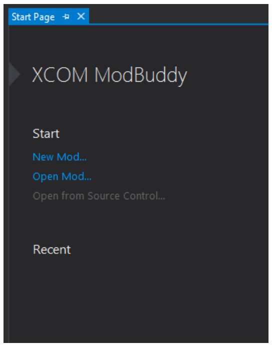
MSBuild project build output verbosity:
Normal

MSBuild project build log file verbosity:
Normal

Start Page

When the IDE first launches the user will be presented with a start page that presents options for creating new mod projects, opening an existing mod project, and accessing the list of recently opened mod projects.

In this quick start guide we will create a new mod project, so **select "New Mod"**

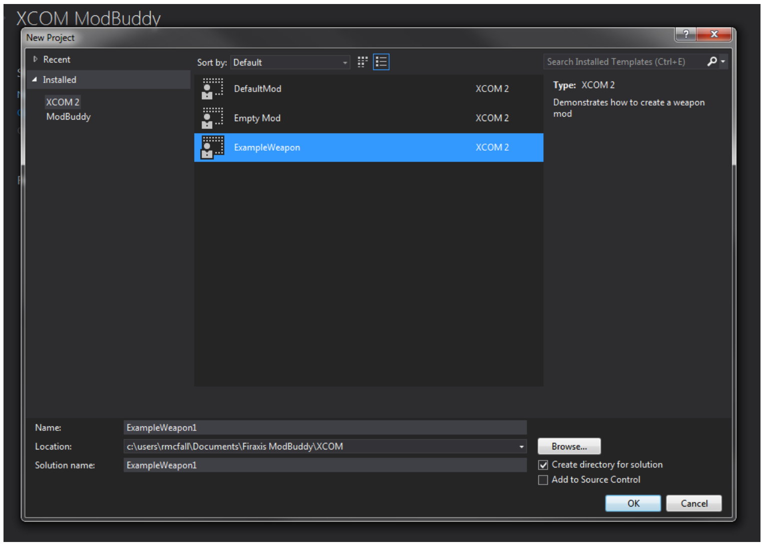


New Project

After selecting "New Mod", the user will be presented with the dialog shown to the right. By default, the dialog shows installed XCOM2 project templates that can be used to create a new project.

In this guide, we will choose the **ExampleWeapon** template project. This project is an example project and so contains all the script code, configuration files, localization files, and content packages necessary to add a new weapon to the game.

Choose ExampleWeapon and select OK.



Type: XCOM 2

Demonstrates how to create a weapon mod

Name: ExampleWeapon1

Location: c:\users\rmcfall\Documents\Firaxis ModBuddy\XCOM

Solution name: ExampleWeapon1

☑ Create directory for solution

□ Add to Source Control

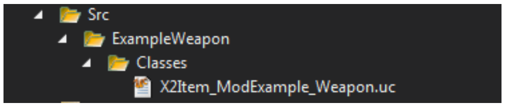
The Mod Pipeline

Now that we have created a mod project, we can review the components that go into a mod as well as build, test, and publish our mod.

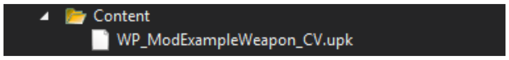
Mod Components

Mods for XCOM2 are composed of many different types of data that combine to form changes to the XCOM 2 base game.

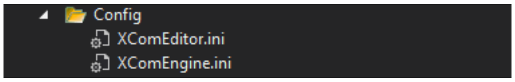
Script Code: Script code is the most powerful type of mod content. In our example weapon mod, the script code defines a **data template** which is the gameplay data describing how the new weapon works. How much damage does it do? Which upgrades can it use? What tech level does it require? Does it need to be built or is it available from the start? How much will it cost? Data templates just scratch the surface for what can be added using script. Script code, if it overrides script that shipped with the game can be used to make a radically different XCOM game.



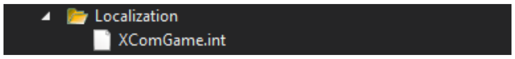
Unreal Content Packages: Unreal content packages can be UPK files or MAP files. These files contain the



content of the game: meshes, textures, animations, sounds, physics models, archetypes, etc. In our example weapon mod, there is a single package called WP_ModExampleWeapon_CV that defines how our new weapon will look, sound, shoot, what projectiles it will use, how it will fall when dropped, etc. This package is referenced by the script **data template** above. Content packages are created through UnrealEd, which will be covered later.



Config Files: Config files, or INIs, are used to provide settings and default values to the game. They are convenient to use because changing them does not require the game to be recompiled, and because the UE3 config file processing system allows for individual config settings to be overridden while leaving the rest unchanged. This handy behavior can be seen in our example weapon mod where XComEditor and XComEngine, both config files with a large amount of settings, simply adds our mod package to the lists that let the engine know our modded script package exists. In this way, many mods making edits to the same config files can co-exist … for the most part.

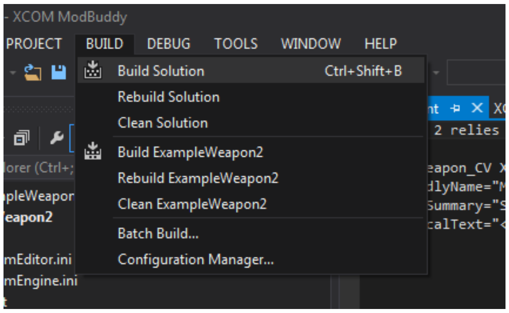


Localization Files: Loc files are really just config files in disguise but they have a specialized purpose: to provide all player facing text in the game. In our example weapon mod, English language (INT) strings are provided to fill out the friendly name and other fields the weapon needs to display correctly in the game.

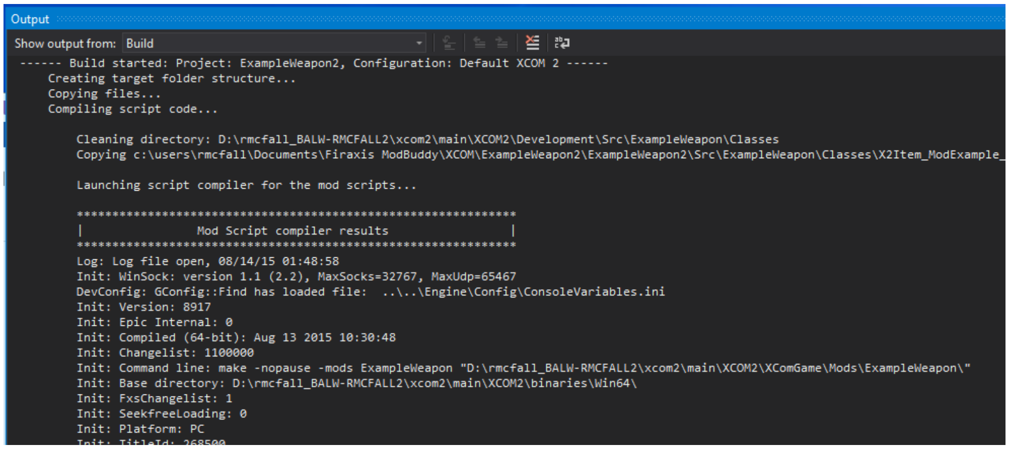
Readme.txt & ModPreview.jpg: The ModPreview image will be used to depict the mod on steam workshop, while the readme.txt can provide instructions or other information related to the mod.

Building

Once the mod content has been created and is ready for testing, it is time to build the mod into a form that the game can load. To do this, use the Build->Build Solution option on the toolbar.



Output Window: After hitting Build Solution, the IDE will display build progress in the output window of the app. During the build process, Modbuddy will create the files necessary to test and upload the mod. Part of this

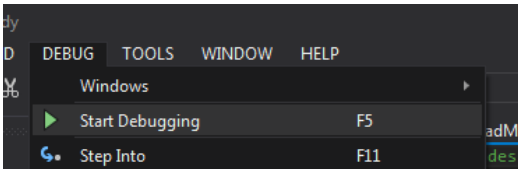


```
------ Build started: Project: ExampleWeapon2, Configuration: Default XCOM 2 ------
    Creating target folder structure...
    Copying files...
    Compiling script code...

        Cleaning directory: D:\rmcfall_BALW-RMCFALL2\xcom2\main\XCOM2\Development\Src\ExampleWeapon\Classes
        Copying c:\users\rmcfall\Documents\Firaxis ModBuddy\XCOM\ExampleWeapon2\ExampleWeapon2\Src\ExampleWeapon\Classes\X2Item_ModExample_

        Launching script compiler for the mod scripts...

        **************************************************************
        |                 Mod Script compiler results                |
        **************************************************************
        Log: Log file open, 08/14/15 01:48:58
        Init: WinSock: version 1.1 (2.2), MaxSocks=32767, MaxUdp=65467
        DevConfig: GConfig::Find has loaded file:  ..\..\Engine\Config\ConsoleVariables.ini
        Init: Version: 8917
        Init: Epic Internal: 0
        Init: Compiled (64-bit): Aug 13 2015 10:30:48
        Init: Changelist: 1100000
        Init: Command line: make -nopause -mods ExampleWeapon "D:\rmcfall_BALW-RMCFALL2\xcom2\main\XCOM2\XComGame\Mods\ExampleWeapon\"
        Init: Base directory: D:\rmcfall_BALW-RMCFALL2\xcom2\main\XCOM2\binaries\Win64\
        Init: FxsChangelist: 1
        Init: SeekfreeLoading: 0
        Init: Platform: PC
        Init: TitleId: 268500
```

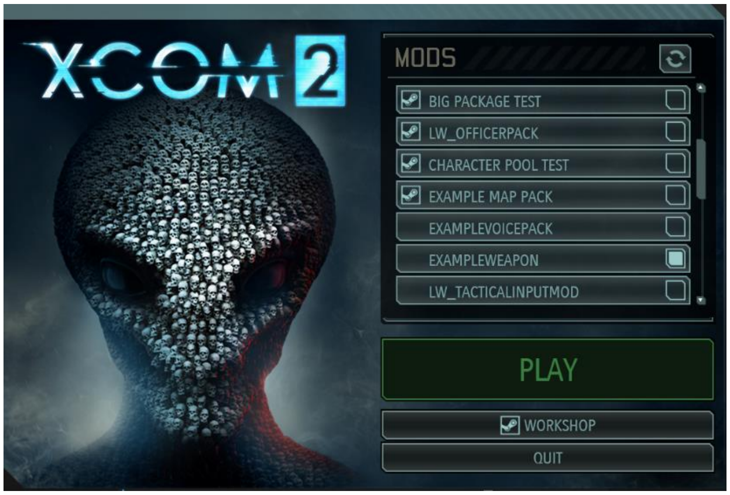
process is the compilation of script code. Be sure to check the script compilation results for success or error codes.

Testing

While developing a mod, it is recommended to test the mod out locally before making it available on steam workshop. The XCOM2 SDK provides a mechanism to do this through the DEBUG menu.



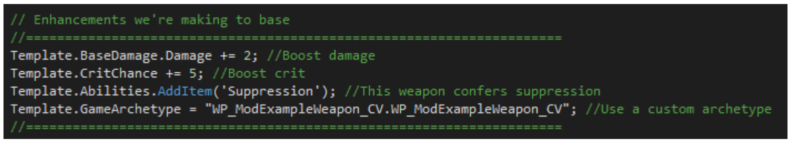
Click the Start Debugging menu option under the DEBUG menu to begin the testing process. This will build the mod, and then start the XCOM 2 launcher.



Select the "ExampleWeapon" mod and hit Play. The launcher will enumerate all mods that are subscribed or installed by the user. Local, or stand-alone mods will have their relative path in the description field. Hitting play will launch XCOM2 with the ExampleWeapon mod enabled.

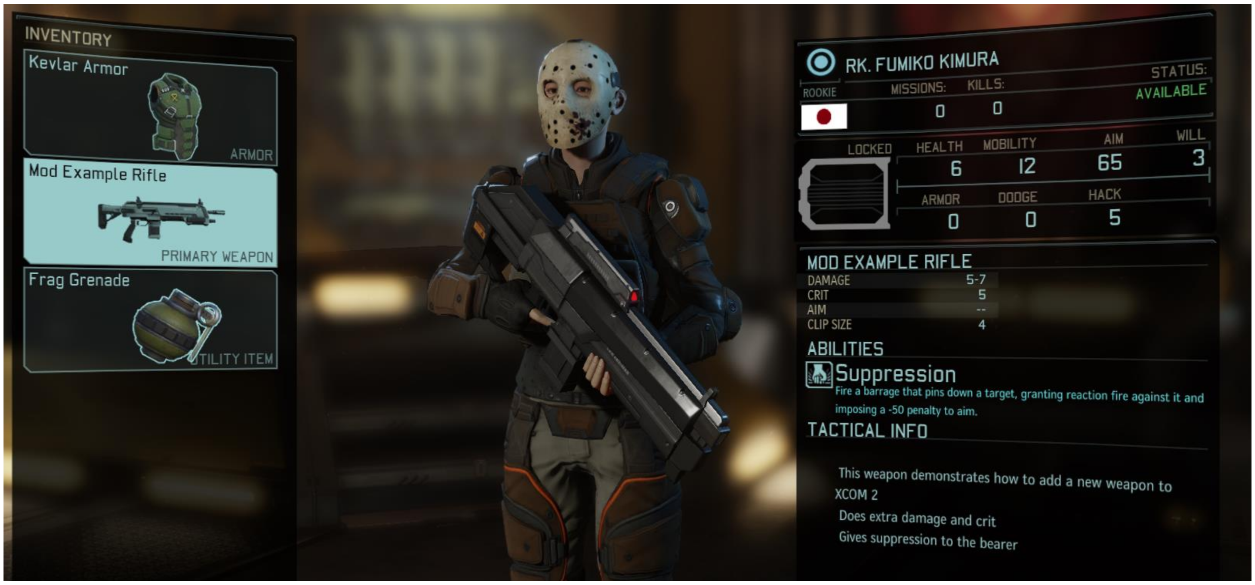
Verify that the mod is operating correctly in the game. The ExampleWeapon mod adds a new weapon that is available from the beginning of the game, so to test it we will launch a single player campaign and equip a soldier with the new weapon.

Here is the script code from the template defining the properties we want to see on the weapon:



```
// Enhancements we're making to base
//======================================================================
Template.BaseDamage.Damage += 2; //Boost damage
Template.CritChance += 5; //Boost crit
Template.Abilities.AddItem('Suppression'); //This weapon confers suppression
Template.GameArchetype = "WP_ModExampleWeapon_CV.WP_ModExampleWeapon_CV"; //Use a custom archetype
//======================================================================
```

And the mod as running in XCOM2:



Fire a barrage that pins down a target, granting reaction fire against it and imposing a -50 penalty to aim.

This weapon demonstrates how to add a new weapon to XCOM 2
Does extra damage and crit
Gives suppression to the bearer

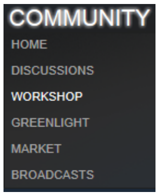
Now that we have confirmed the mod operates as intended, we are ready to publish it.

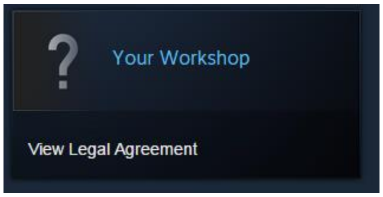
Publishing

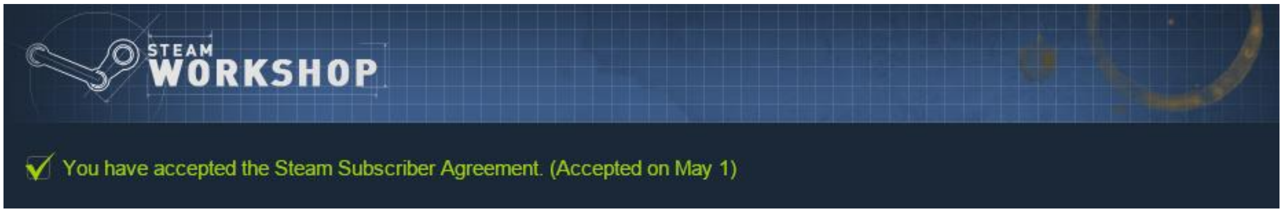
Mods can be published through Steam Workshop or distributed as stand-alone bundles.

Publishing to Steam Workshop

To publish to the Steam Workshop, first visit the Steam Workshop page and verify that your account has agreed to the license agreement there**.** This can be found under Community->Workshop->Your Files->View Legal Agreement.

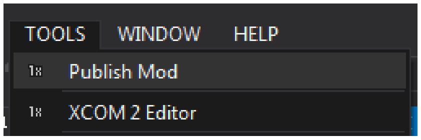




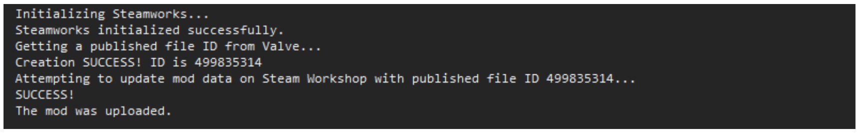


✓ You have accepted the Steam Subscriber Agreement. (Accepted on May 1)

Next, use the Tools->Publish menu option available in the XCOM2 IDE. This will launch the steam workshop UGC upload process.



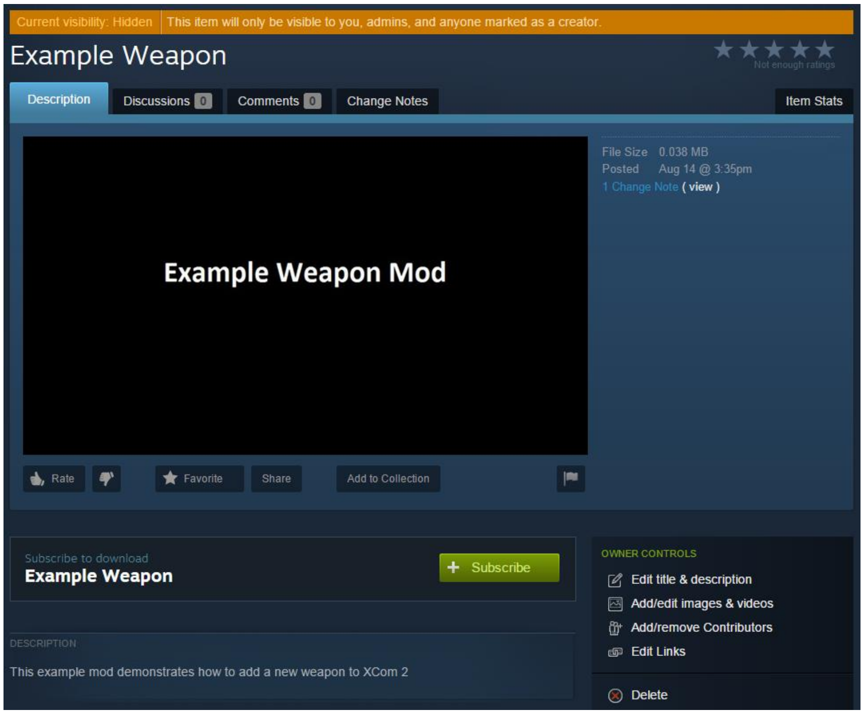
Information regarding the workshop item creation and uploading process will be reported to the output window of the IDE.



```
Initializing Steamworks...
Steamworks initialized successfully.
Getting a published file ID from Valve...
Creation SUCCESS! ID is 499835314
Attempting to update mod data on Steam Workshop with published file ID 499835314...
SUCCESS!
The mod was uploaded.
```

Visit your user's Steam Workshop item page to view the published mod.

Note that the first time a mod is uploaded its visibility will default to Hidden. In order to make the mod available publicly the visibility of the mod item will need to be set.



Current visibility: Hidden | This item will only be visible to you, admins, and anyone marked as a creator.

Example Weapon

File Size 0.038 MB
Posted Aug 14 @ 3:35pm
1 Change Note (view)

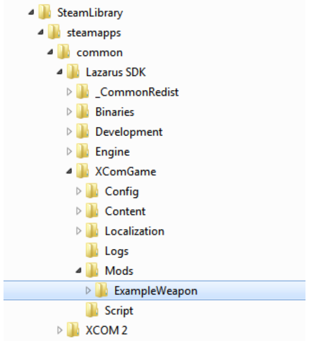
Subscribe to download
Example Weapon

DESCRIPTION
This example mod demonstrates how to add a new weapon to XCom 2

Publishing Stand-alone

Publishing stand-alone mods involves bundling up the mod content for distribution through a non-steam source.

To publish a stand-alone mod, locate the XCOM2 SDK base directory, and the XComGame\Mods folder within it. It will typically be contained within a steam library or steam installation folder – an example is shown to the right.



Zip up / package the ExampleWeapon folder for redistribution.

All that is necessary to install the mod on a user's machine is to unpack the ExampleWeapon folder into the same XComGame\Mods location in their installation of XCOM2.