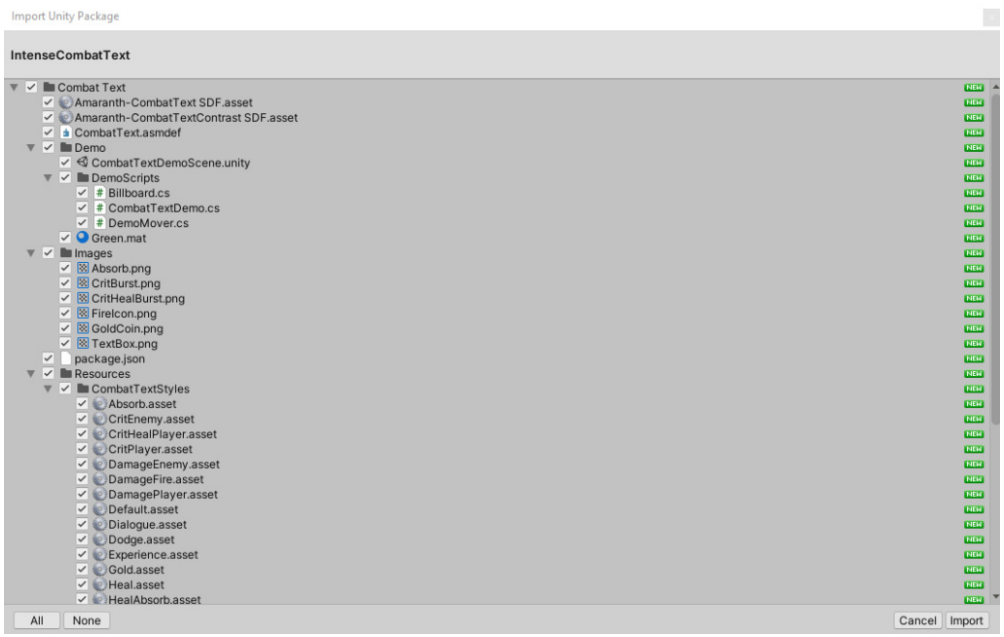
Intense Documentation

The latest version of this documentation can always be found at this link. (https://devplz.com/combat-text-documentation/)

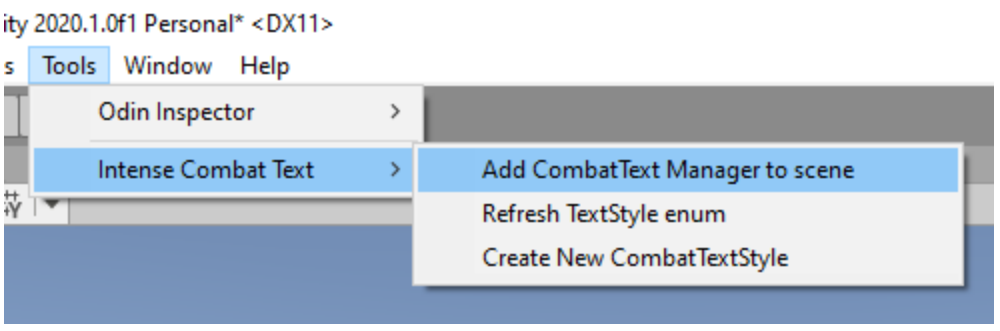
Quick Start Guide

Step 1: Import the Insane Combat Text Package. If you're reading the PDF version of this, congrats, you're ahead of the game!

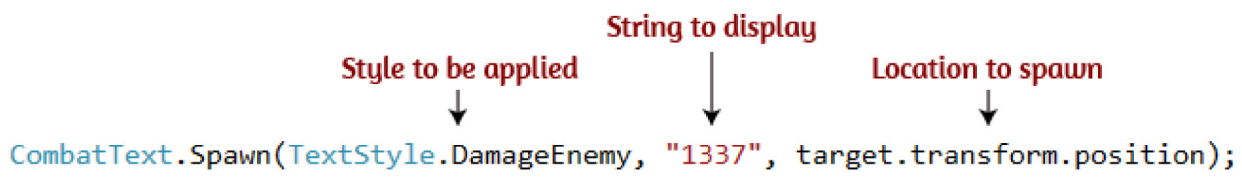
Note: This asset **requires** the latest "Verified" TextMeshPro package. **Be sure to import TMP before this.**



Step 2: From the Unity Editor, click "Tools/Intense Combat Text/Add Combat Text Manager to scene".



Step 3: Call CombatText.Spawn() somewhere in your code. That's it!



Combat Text Manager

The Combat Text Manager prefab needs to be present in a scene in order to call CombatText.Spawn(). It can be added from the Tools/Intense Combat Text menu and requires no set up to function.

Alternatively, if you don't want to use the prefab, you can just add a CombatText.cs component to any of your existing scene objects.

CombatText Component Reference

Show Text- When set to False, no CombatText will be displayed. If you want to hook it up to your Settings menu, call CombatText.Show() to enable or disable it.

DontDestroyOnSceneChange- If True, this GameObject will persist between scenes. Especially useful if you use a bootstrapper scene.

Style Asset Path- The path relative to a Resources folder. Only CombatTextStyle assets at this path will be loaded. Right click the component and choose Reset to set the default path.

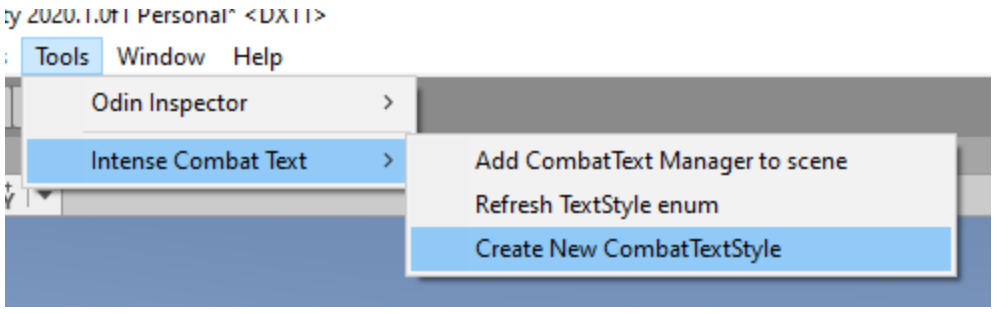
Text Prefab- The default CombatTextInstance prefab to use. Most users can leave it at the default. However, if for whatever reason you need to make significant changes to the default prefab, it is recommended that you create a Prefab Variant of the default prefab and add that here.

Customizing Text Styles

"CombatTextStyle" assets are used to determine how a specific TextStyle looks and moves. Intense Combat Text includes quite a few example

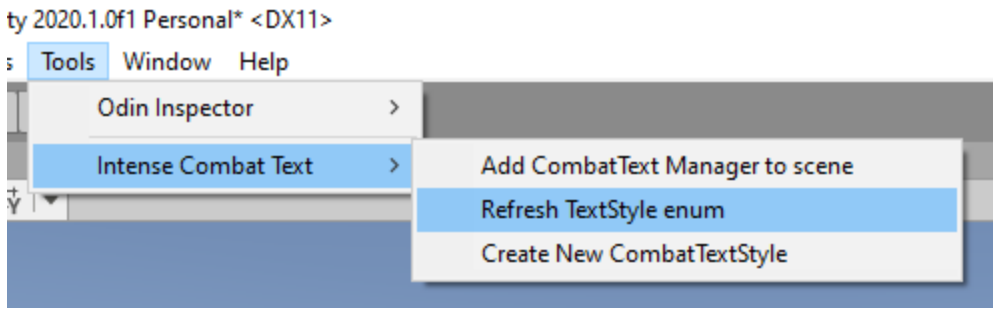
styles to get you started, but if you want to make your own, I promise not to take it personally.

Creating a new CombatTextStyle asset



Step 1- In the Unity Editor, click “Tools/Intense Combat Text/Create New CombatText Style”. A new CombatTextStyle asset will be created in the Assets/Intense Combat Text/Resources/CombatTextStyles folder.

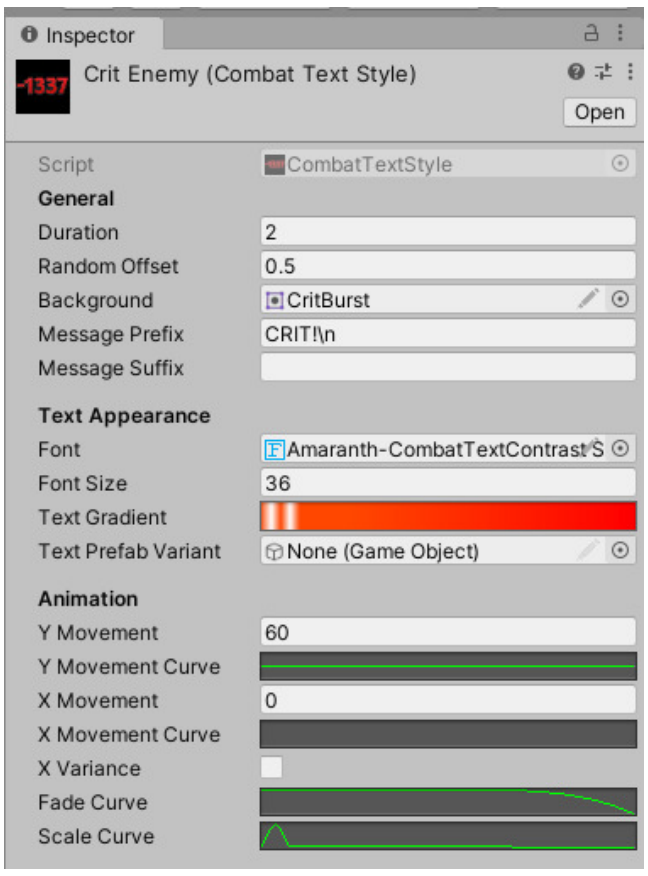
Rename the new file.



Step 2- Now click “Tools/Intense Combat Text/Refresh TextStyle enum” and Wait for Unity to finish compiling.

CONGRATS! You can now reference your new Style in CombatText.Spawn() using TextStyle.YourNewStyleName.

CombatTextStyle Property Reference



Duration- The time from spawn until the new text automatically despawns.

Random Offset- A random offset from the spawn location on the X axis. Example: Using an offset of 1 will offset it randomly between -1 and 1.

Background- Optional UI sprite to display behind the text. Supports regular and 9 sliced sprites.

Message Prefix & Suffix- Optional text to always include before/after the main text. Can include Rich Text tags (http://digitalnativestudios.com/textmeshpro/docs/rich-text/) and inline Sprites. (http://digitalnativestudios.com/textmeshpro/docs/sprites/) (See “Gold,” “Absorb,” and “Fire Damage” styles for examples.)

Font- The TextMesh Pro Font Asset (https://learn.unity.com/tutorial/textmesh-pro-font-asset-creation) to use for the text.

Font Size- I mean..

Text Gradient- The color of the text. If a gradient is used, it will animate its color from left to right over the duration

Text Prefab Variant- Optional prefab variant to use instead of the default CombatTextInstance Prefab. Only needed if some of your styles need to be significantly different from others.

X & Y Movement- Controls the text’s x/y position over its duration, multiplied by the x/yMovementCurve.

X & Y Movement Curve- The text’s x/y movement curve over its duration, multiplied by the x/yMovement.

X Variance- When true, instead of xMovement being a static value, each newly spawned text’s xMovement will choose a random value being xMovement and -xMovement.

Fade Curve- Controls the alpha value of the text over its duration. 1 for full opacity, 0 for fully transparent.

Scale Curve- Controls the scale of the text over the duration.

Intense Pro Tip- When you want to create a new TextStyle with similar animation/etc to an existing style, you can save time by duplicating it. Rename the duplicate and make your changes. Note that you can also copy/paste AnimationCurves and Gradients by right clicking.

Combat Text Follow Targets

CombatText.Spawn() can be passed a Transform component as an optional parameter (followTarget); the spawned CombatText will now follow it. Note that rather than repositioning itself every frame, to save performance the spawned CombatTextInstance will become a child of the specified follow target.

As a result, if the follow target is destroyed, any CombatText that is a child of it will also be destroyed. While this won’t cause any errors, it may cause undesirable behavior. For instance, you attack an enemy and kill him, but the CombatText that should show the damage dealt is instantly destroyed along with him. Fortunately, there are two ways around this.

Option 1- Add a CombatTextDetacher component to any GameObject that will be used as a follow target, and any CombatText will automatically persist past its destruction. It’s recommended to use a child of your Character/Enemy GameObject as the follow target, this way the child’s local position can be changed to offset the CombatText spawn point. The “CombatTextFollowTarget” prefab is included for this purpose.

Option 2- The CombatText.DetachAllFromFollowTarget() method can be called within the the to-be-destroyed object’s OnDisable() or OnDestroy() methods. Simply pass in the FollowTarget’s transform.

Troubleshooting

I'm calling CombatText.Spawn() but no text is spawning

Be sure the CombatText Manager prefab is somewhere in your scene, or a CombatText.cs component is on another scene object.

I created a new CombatTextStyle asset but it's not showing up in the TextStyle enum

1. Every time you create a new Text Style, you have to rebuild the enum file by clicking Tools/Intense Combat Text/Refresh TextStyle enum.

2. All CombatTextStyle assets must be located in a subfolder of a “Resources” folder, and that folder has to match the Style Asset Path property on the scene’s CombatText Manager.

I'm getting "The type or namespace name 'TMPro' could not be found' errors

Be sure to have the most recent “Verified” TextMesh Pro package imported into your project. “Preview” versions have been known to cause issues.

I'm getting the following error: MissingComponentException: There is no 'CanvasRenderer' attached to the "TMP UI SubObject [TextMeshPro/Sprite]" game object, but a script is trying to access it.

This happens if you have the TextMesh Pro 3.0.0-preview.1 package installed and you try to use an inline sprite. This can be corrected by updating your TextMesh Pro package to the latest Verified version. (3.0.1 at the time of this writing.)

I'm having an issue not listed here!

Please send an email to support@devplz.com. Be sure to describe your problem in as much detail as possible, and copy and paste the full stack trace of any errors you’re seeing. Please be patient, I try to answer as fast as I can, but sometimes it can take a few days to to respond.