

Getting Started - Emotion Card Util

This Util lets you load Emotion Cards while using a Packageld and a number Id, so even if 2 mods use the same number Id, they won't clash with each other and this Mod is used with XML files.

Discord

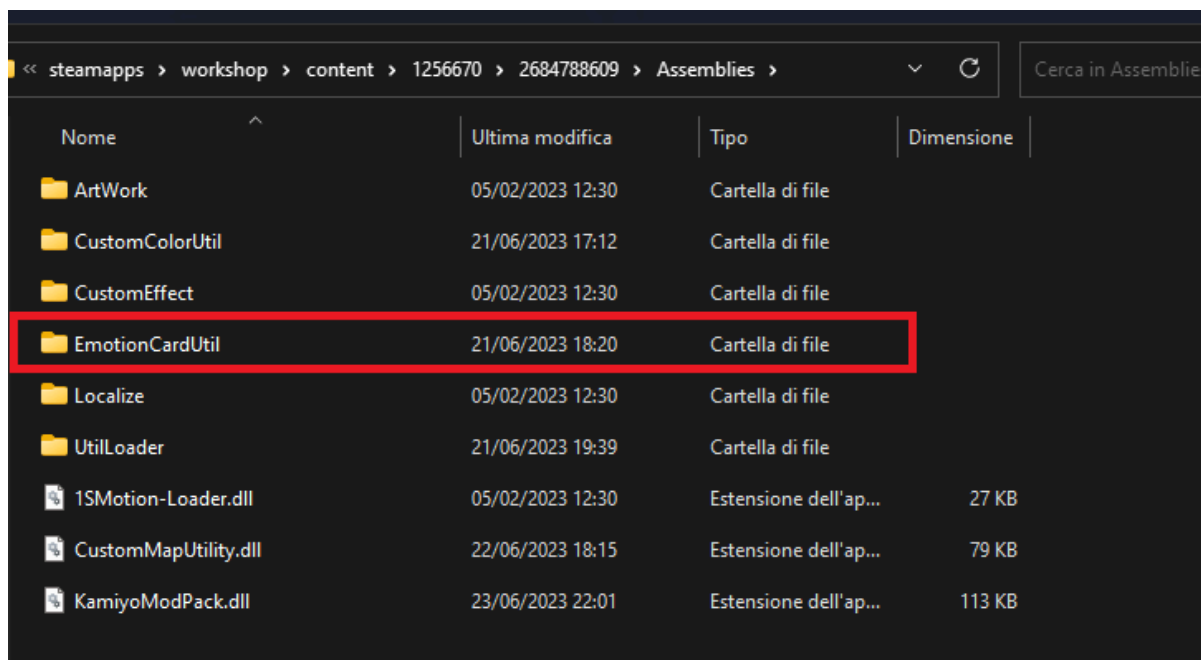
If you have any questions or you want help for some steps here, you can write to me on Discord. My Discord IGN is kamiyo

Created a Discord Server for who needs help with the Util or for who wants to report a bug.
Discord Server [Link](#)

How to use the Util

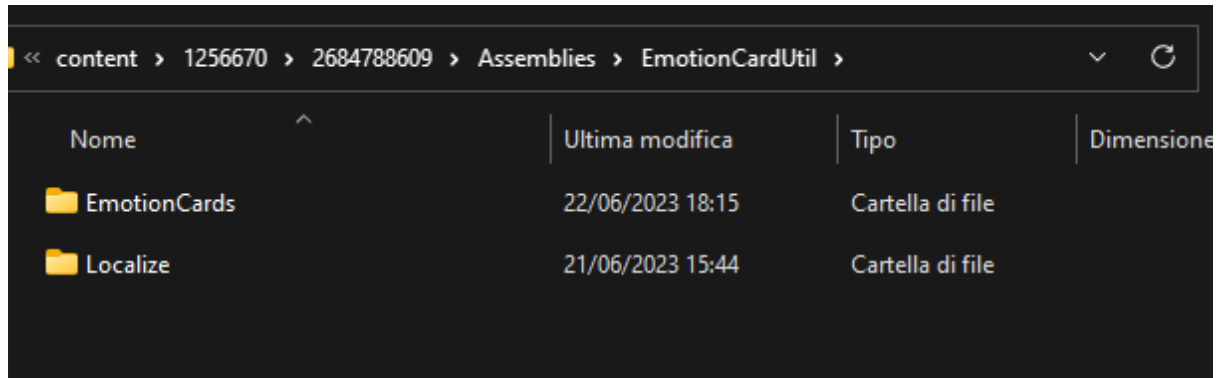
This is a guide for a basic use of the Util, the util can do more things but they are more advanced and in most cases they are not really needed.

Assemblies Folder



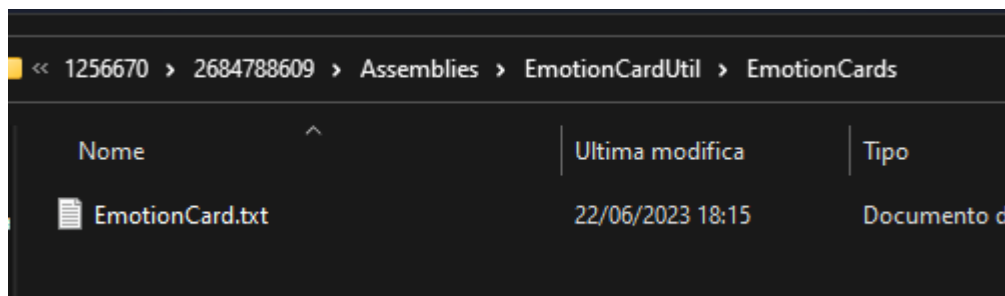
In the Assemblies Folder of the Mod, we create a EmotionCardUtil folder, it will tell the Util that it should load Emotion Cards from the mod.

EmotionCards and Localize



After creating the first folder, we'll create those 2 folders, one is the core of the Emotion Cards, the other one is the localization of what we see in game.

EmotionCards



We can create the file here (txt or xml) that will contain the core of our Emotion Cards.

```
<?xml version="1.0" encoding="utf-8"?>
<EmotionCardXmlRootExtension xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <!-- EmotionCardXmlInfo -->
  <EmotionCard ID="21341">
    <Name>OldSamurai1_21341</Name>
    <Artwork>OldSamurai1_21341.png</Artwork>
    <State>Positive</State>
    <Sephirah>None</Sephirah>
    <Level>2</Level>
    <EmotionLevel>2</EmotionLevel>
    <EmotionRate>0</EmotionRate>
    <Locked>>false</Locked>
    <TargetType>SelectOne</TargetType>
    <Script>OldSamurai1_21341</Script>
  </EmotionCard>
</EmotionCardXmlRootExtension>
```

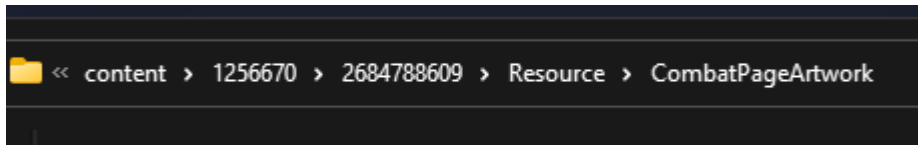
The root of this file is called **EmotionCardXmlRootExtension** (full line
<EmotionCardXmlRootExtension xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">)

XML parameters :

Id (Attribute of EmotionCard) : The Id of the card.

Name: Is the Id of the localization where it will take the Name and the Description of the Emotion Card.

Artwork : The name of the Card image as png found in the base **CombatPageArtwork** folder of the mod.



State: Here we write if the Card is Positive or Negative.

Sephirah: Set the Sephirah of the Card (it should be always None)

Level: The level of the emotion Card (value can change between 1-3).

Locked : If the card is not usable, we can leave this to false since we can do logic in our mod itself on how to use it or unlock it.

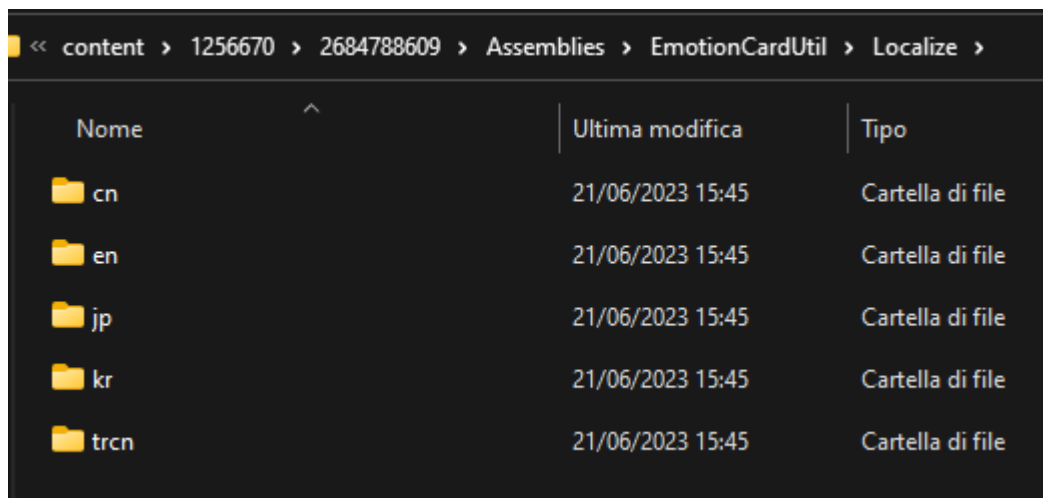
TargetType: Here we write if the target is a Single Target, All Units (only allies) or All Units including Enemies. ([SelectOne](#) / [All](#) / [AllIncludingEnemy](#))

Script: The Script of the Emotion Card. It is similar NOT EQUAL to the PassiveAbilityBase and is named **EmotionCardAbilityBase** (as in the example picture)

```
0 references
public class EmotionCardAbility_Kamiyo3_21341 : EmotionCardAbilityBase
{
    0 references
    public override void OnRoundStart()
    {
        foreach (var unit in BattleObjectManager.instance.GetAliveList().Where(x:BattleUnitModel => x != _owner))
            unit.AddBuff<BattleUnitBuf_AlterEnergy_Re21341>(stack: 1, maxStack: 10);
    }

    0 references
    public override void OnWinParrying(BattleDiceBehavior behavior)
    {
        var target:BattleUnitModel = behavior.card?.target;
        if (target == null) return;
        _owner.SetEmotionCombatLog(_emotionCard);
        target.AddBuff<BattleUnitBuf_AlterEnergy_Re21341>(stack: 1);
        if (target.GetActiveBuff<BattleUnitBuf_AlterEnergy_Re21341>() != null) _owner.breakDetail.RecoverBreak(2)
    }
}
```

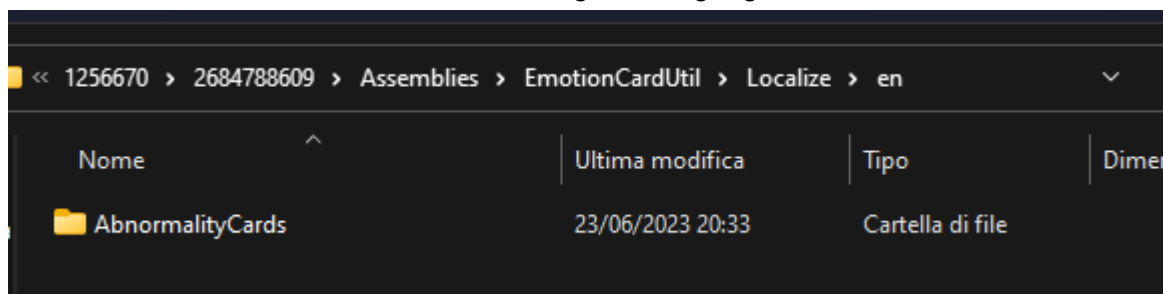
Localize



A screenshot of a file explorer window showing the path: <code><< content > 1256670 > 2684788609 > Assemblies > EmotionCardUtil > Localize ></code>. The main area displays a table of folders:

Nome	Ultima modifica	Tipo
cn	21/06/2023 15:45	Cartella di file
en	21/06/2023 15:45	Cartella di file
jp	21/06/2023 15:45	Cartella di file
kr	21/06/2023 15:45	Cartella di file
trcn	21/06/2023 15:45	Cartella di file

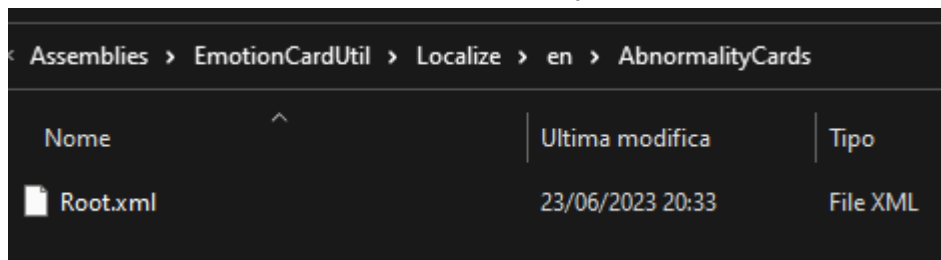
Here we have the localization based on the game languages.



A screenshot of a file explorer window showing the path: <code><< 1256670 > 2684788609 > Assemblies > EmotionCardUtil > Localize > en</code>. The main area displays a table of folders:

Nome	Ultima modifica	Tipo	Dimensione
AbnormalityCards	23/06/2023 20:33	Cartella di file	

And inside we'll create the folder AbnormalityCards.



A screenshot of a file explorer window showing the path: <code><< Assemblies > EmotionCardUtil > Localize > en > AbnormalityCards</code>. The main area displays a table of files:

Nome	Ultima modifica	Tipo
Root.xml	23/06/2023 20:33	File XML

Then we create a file inside (Xml or txt) where we'll put the data of the emotion Cards that we need.

```
<?xml version="1.0" encoding="utf-8"?>
<AbnormalityCardsRoot xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <Sephirah SephirahType="ETC">
  <AbnormalityCard ID="OldSamurai1_21341">
    <Abnormality>Old Samurai</Abnormality>
    <CardName>Duel</CardName>
    <AbilityDesc>[Single Ally] At the start of each Scene, add a Counter Die (Slash 4-8) to the queue. If Powernu
    <FlavorText>This is the way of a Samurai. There is no ending in fighting.</FlavorText>
    <Dialogues>
      <Dialogue ID="1">With this sword, Let's end it!</Dialogue>
      <Dialogue ID="2">Now it's you and me. Let's go!</Dialogue>
      <Dialogue ID="3">I'll fight until the end, even if there is never an end on the way of a Samurai.</Dialogue>
    </Dialogues>
  </AbnormalityCard>
  </AbnormalityCardsRoot>
```

The root of the file is called **AbnormalityCardsRoot** (full line : <code><AbnormalityCardsRoot xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema"></code>)

First we create the node **Sephirah** with the attribute **SephirahType="ETC"**, we'll need to close this Node only after writing **ALL** the cards, not just one.

Then we create the node **AbnormalityCard** with the **Id Attribute** equal to the **Name** we wrote in the other EmotionCards file.

Abnormality : The owner of the emotion cards, it groups the emotion cards that it should be of the same "Character"



CardName : As the word says, it is the name of the Card

AbilityDesc : The description of the Card, remember to close the node at the end.



FlavorText : The text under the picture in the Card.

Dialogues : It will contain **Dialogue** nodes with an attribute Id that is just an order number and Text that will appear on our units when the emotion card is applied on them.

Floor Ego Cards

For using floor Ego cards we need to do a few things, first we must open the **CardInfo.xml** in the **Data** folder of our mod.

IMPORTANT NOTE : For those who use the PM Invitation tool, once you put an option on a Card, the tool will not see the cards anymore, so saving in the tool, without putting read-only on CardInfo.xml WILL WIPE ALL THE CARDS.

Steam > steamapps > workshop > content > 1256670 > 2684788609

Nome	Ultima modifica	Tipo
Assemblies	23/06/2023 22:01	Cartella di
Data	23/06/2023 20:33	Cartella di
Resource	05/02/2023 12:30	Cartella di
StageModInfo.xml	26/06/2023 08:49	File XML
UploadID.xml	26/06/2023 08:49	File XML

<< Steam > steamapps > workshop > content > 1256670 > 2684788609 > Data

Nome	Ultima modifica	Tipo	Dimensione
StoryEffect	05/02/2023 12:30	Cartella di file	
StoryText	05/02/2023 12:30	Cartella di file	
BookStory.xml	26/06/2023 08:49	File XML	3 KB
CardDropTable.xml	26/06/2023 08:49	File XML	2 KB
CardInfo.xml	26/06/2023 08:49	File XML	67 KB
Combat_Dialog.xml	26/06/2023 08:49	File XML	2 KB
Deck_Enemy.xml	26/06/2023 08:49	File XML	3 KB
Dropbook.xml	26/06/2023 08:49	File XML	3 KB
EnemyUnitInfo.xml	26/06/2023 08:49	File XML	5 KB
EquipPage_Enemy.xml	26/06/2023 08:49	File XML	11 KB
EquipPage_Librarian.xml	26/06/2023 08:49	File XML	24 KB
PassiveList.xml	26/06/2023 08:49	File XML	26 KB
StageInfo.xml	26/06/2023 08:49	File XML	7 KB

After it we choose the card that we did, and we add `<Option>EGO</Option>` node

```

<Card ID="900">
  <Name>Sakura Mirage</Name>
  <TextId>-1</TextId>
  <Artwork>Mirage.png</Artwork>
  <Option>EGO</Option>
  <Parity>uniques</Parity>
  <Spec Range="FarAreaEach" Cost="0" Affection="Team" EmotionLimit="0" />
  <Script>MioMassAttack_Re21341</Script>
  <ScriptDesc />
  <BehaviourList>
    <Behaviour Min="4" Dice="9" Type="Atk" Detail="Slash" Motion="J" MotionDefault="N" EffectRes="" Script="purpleAreaD
    <Behaviour Min="4" Dice="9" Type="Atk" Detail="Slash" Motion="H" MotionDefault="N" EffectRes="" Script="purpleAreaD
    <Behaviour Min="4" Dice="8" Type="Atk" Detail="Slash" Motion="Z" MotionDefault="N" EffectRes="" Script="purpleAreaD
  </BehaviourList>
  <Chapter>7</Chapter>
  <SpecialEffect />
  <SkinChange />
  <SkinChangeType>EGO</SkinChangeType>
  <SkinHeight>0</SkinHeight>
  <MapChange />
  <Priority>500</Priority>
  <PriorityScript />
  <Category>None</Category>
  <MaxCooltimeForEgo>4</MaxCooltimeForEgo>
  <MaxNum>150</MaxNum>
</Card>

```

And that's all. After it we write this line of code when we want to add it to our Team.

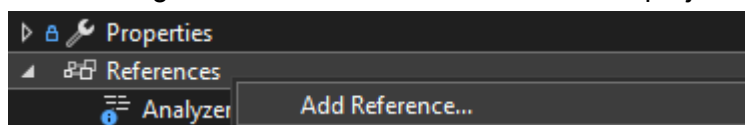
```
Singleton<SpecialCardListModel>.Instance.AddCard(new LorId(packageId: "YourModId", id: 1 /*card number id*/), Singleton<StageController>.Instance.CurrentFloor);
```

**Singleton<SpecialCardListModel>.Instance.AddCard(new
LorId("YourModId",1),Singleton<StageController>.Instance.CurrentFloor);**

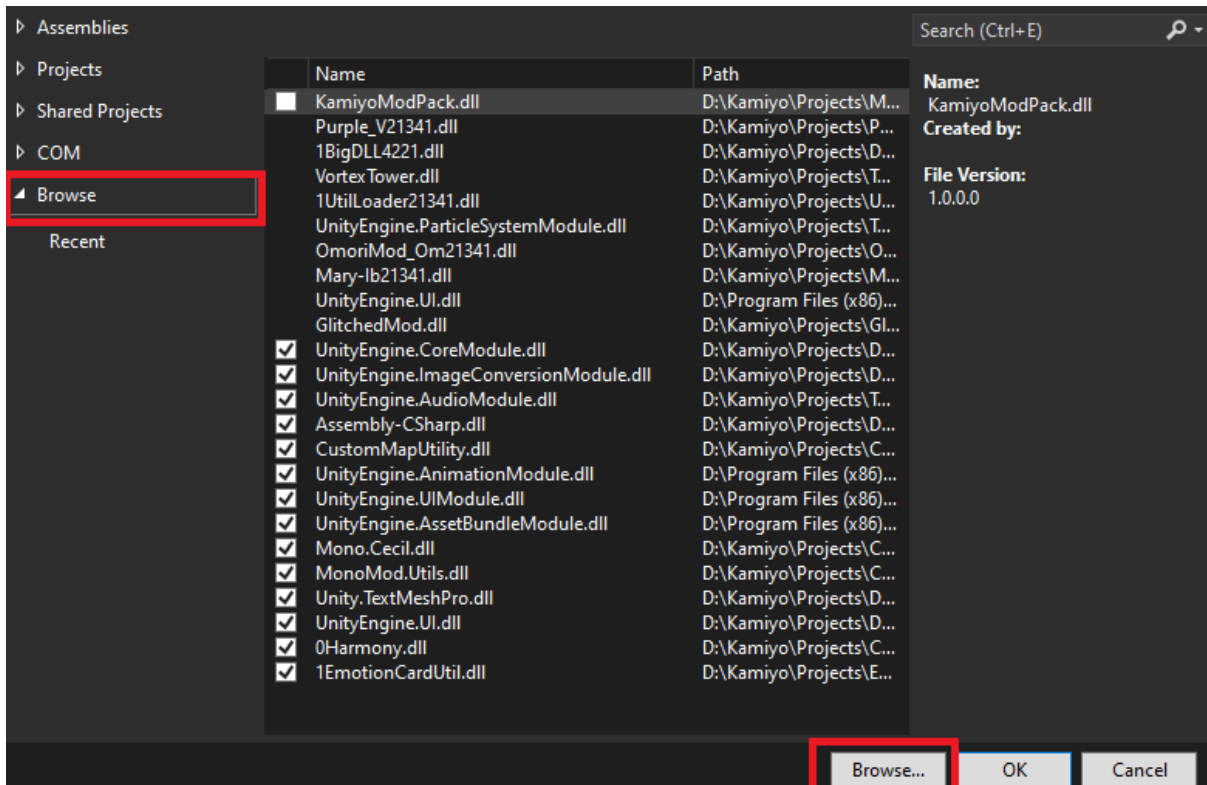
And it will be added to everyone in the team with shared cooldown as a Floor Ego Card.

Util Code

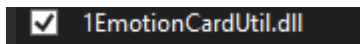
Before using this line of code we must add to the project reference the util dll.



By pressing with the right click of the mouse on **References** and choosing **Add Reference...**



We must search the **1EmotionCardUtil.dll** with the Browse button



and after checking it, we'll press **Ok** and we are ready to use the line of code that we need.

```
owner.emotionDetail.ApplyEmotionCard(EmotionCardLoader.GetEmotionCard(packageId: "MyModId", cardId: 1 /* Number Id of the card */));
```

With this line of the code, we can apply the Emotion Card that we want to our unit, using a **BattleUnitModel (owner in this case)**. We take our custom card with the **EmotionCardLoader.GetEmotionCard(ModId, CardId (int))** method.

Files

If you want to check the files themselves of the screenshots, they are in my mod pack. [Link](#)