

ITpMessaging

ITpMessaging is used to implement targets for the Messaging Service.

It's pretty simple:

```
/// <summary>
/// Interface for using TpNet SendMessage methods.
/// </summary>
/// <typeparam name="T">Type for sending a message</typeparam>
public interface ITpMessaging <in T> where T:MessagePacket<T>
{
    /// <summary>
    /// Send a message of type T
    /// </summary>
    /// <param name="sentPacket">The sent packet.</param>
    void MessageTarget(T sentPacket);

    /// <summary>
    /// Optional "are you ready?" method that can be used in filtering
    /// prior to sending a message. Useful in some edge cases. Override
    /// in implementation if necc. NOTE this is NOT checked internally
    /// somehow. You have to use a filter and test this.
    /// </summary>
    /// <returns>True if the tile is prepared to get the message.</returns>
    bool CanAcceptMessage() { return true; }
}
```

Message Packets are discussed [here](#).

In your custom tile code you use explicit implementations for these members, for example:

```
void ITpMessaging<ActionToTilePacket>.MessageTarget(ActionToTilePacket sentPacket)
{
    ActivateAnimation(!AnimationIsRunning);
}
```

`T` is `ActionToTilePacket`. In this example, the packet information is ignored.

Here's a more complex example where the packet contents are used to control what happens. In this case, `T` is `PositionPacket`, which just contains a position (like the Player position). This is from `TpAnimZoneSpawner`.

```
void ITpMessaging<PositionPacket>.MessageTarget(PositionPacket sentPacket)

{
    var pos = sentPacket.m_Position;
    lastContactPosition = pos;

    pos -= m_TileGridPosition; //remove offset
    if (m_ZoneBoundsInt.Contains(pos))
        SpawnTileOrPrefab();
}
```

This tile has a 'Zone' (aka, `BoundsInt`) that describes an area. It doesn't even have to cover the tile itself. But the `BoundsInt` position is the offset of the zone from the tile's position and not an absolute position: think of it as relative addressing.

That's why the tile's grid position is subtracted from the position information in the packet before seeing if the packet's position information is a point within the `BoundsInt`.

If the position is within the `BoundsInt` then we spawn something, based on how the tile is set up.

Revision #3

Created 2025-07-12 12:57:08 UTC by Vonchor

Updated 2025-07-12 13:15:59 UTC by Vonchor