Scripting API

AzureSkyManager.SetNewWeatherProfile()

```
public void SetNewWeatherProfile(int index)
```

Changes the global weather with a smooth transition.

index: The target profile number in the "Global Weather Profiles" list. Set the index to -1 if you want to reset the global weather back to the default day profile.

AzureSkyManager.SetNewDayProfile()

```
public void SetNewDayProfile(AzureSkyProfile profile, float transitionTime)
```

Changes the current day profile with transition.

profile: The new profile.

transitionTime: The time in seconds the transition to the new profile will last.

```
public void SetNewDayProfile(AzureSkyProfile profile)
```

Changes the current day profile without transition.

profile: The new profile.

AzureSkyManager.timeController.GetDayOfWeek()

```
public int GetDayOfWeek (int year, int month, int day)
```

Get the day of the week from a custom date and return an integer between 0 and 6.

AzureSkyManager.timeController.GetCurrentDayOfWeek()

```
public int GetCurrentDayOfWeek ()
```

Get the day of the week from the current date and return an integer between 0 and 6.

AzureSkyManager.timeController.UpdateCalendar()

```
public void UpdateCalendar ()
```

Adjust the calendar when there is a change in the date. Always call this after changing the date by script.

AzureSkyManager.timeController.StartNextDay()

```
public void StartNextDay ()
```

Starts the next day based on the repeat mode selected in the Inspector.

AzureSkyManager.settings

```
public AzureSkySettings settings = new AzureSkySettings();
```

Stores the current sky settings after performing all profile blend transitions.

AzureSkyManager.eventSystemList

```
public List<AzureEventAction> eventSystemList = new List<AzureEventAction>();
```

List used to store all Event Actions created by the user.

AzureSkyManager.outputSystemList

```
public List<AzureOutput> outputSystemList = new List<AzureOutput>();
```

List used to store all outputs created by the user.

How to use:

```
mySliderOutput = AzureSkyManager.outputSystemList[index].floatOutput;

myColorOutput = AzureSkyManager.outputSystemList[index].colorOutput;

myVector2Output = AzureSkyManager.outputSystemList[index].vector2Output;

myVector3Output = AzureSkyManager.outputSystemList[index].vector3Output;
```

AzureSkyEffects.InstantiateThunderEffect()

```
public void InstantiateThunderEffect(int index)
```

Create a thunder effect in the scene using the settings and position from the “Thunder List”. When the thunder sound is over, the instance is automatically deleted.

```
public void InstantiateThunderEffect(int index, Vector3 worldPos)
```

Create a thunder effect in the scene using the settingsfrom the “Thunder List”. When the thunder sound is over, the instance is automatically deleted.

Making a Timeline Transition

There are several variations of the same method to make it easier to create a timeline transition (fast progression from one time of day to another).

AzureSkyManager.StartTimelineTransition()

```
public void StartTimelineTransition(float source, float destination, float transitionTime)
```

Starts a timeline transition from source time of day to a destination time of day that last a period of time in seconds.

```
public void StartTimelineTransition(float destination, float transitionTime)
```

The same as above, but the source is automatically set by taking the current time of day.

```
public void StartTimelineTransition(float destination)
```

This variation contains only one parameter (the destination time of day), this is ideal for use with Unity's events that only allow us to use methods with one parameter, the source is automatically set by taking the current time of day. Since the transition time is not set, you must first call the `public void SetTimelineTransitionTime(float transitionTime)` method to set the transition time duration.

```
public void StartTimelineTransition()
```

This variation starts a timeline transition, but does not set any parameters. You must first call the following methods for everything to work properly.

```
public void SetTimelineSourceTransitionTime(float source)
```

```
public void SetTimelineDestinationTransitionTime(float destination)
```

```
public void SetTimelineTransitionTime(float transitionTime)
```