MIGRATION

Chronos is designed integrate into any project with as few changes as possible. Unfortunately, there are some cases where you will have to make minimal changes to your scripts. For your convenience, these are all catalogued in the following tables.

⏳ Time

Instead of ...	Use...
`Time.deltaTime`	`Timeline.deltaTime`
`Time.fixedDeltaTime`	`Timeline.fixedDeltaTime`
`Time.timeScale`	`Timekeeper.Clock("Root").localTimeScale`
`new WaitForSeconds()`	`Timeline.WaitForSeconds()`

🎬 Animation

Instead of ...	Use...
`Animator.speed`	`Timeline.animator.speed`
`AnimationState.speed`	`Timeline.animation.speed`

Particles

Instead of ...	Use...
`ParticleSystem.playbackSpeed`	`Timeline.particleSystem.playbackSpeed`
`ParticleSystem.time`	`Timeline.particleSystem.time`

`ParticleSystem.isPlaying`
`ParticleSystem.isPaused`
`ParticleSystem.isStopped`
	`Timeline.particleSystem.isPlaying`
	`Timeline.particleSystem.isPaused`
	`Timeline.particleSystem.isStopped`

`ParticleSystem.Play()`
`ParticleSystem.Pause()`
`ParticleSystem.Stop()`
	`Timeline.particleSystem.Play()`
	`Timeline.particleSystem.Pause()`
	`Timeline.particleSystem.Stop()`

Audio

Instead of ...	Use...
`AudioSource.pitch`	`Timeline.audioSource.pitch`

Navigation

Instead of ...	Use...
`NavMeshAgent.speed`	`Timeline.navMeshAgent.speed`
`NavMeshAgent.angularSpeed`	`Timeline.navMeshAgent.angularSpeed`

Physics

Unless specified, all members below have a `Timeline.rigidbody2D` equivalent for 2D.

Instead of ...	Use...
`Rigidbody.mass`	`Timeline.rigidbody.mass`
`Rigidbody.velocity`	`Timeline.rigidbody.velocity`
`Rigidbody.angularVelocity`	`Timeline.rigidbody.angularVelocity`
`Rigidbody.drag`	`Timeline.rigidbody.drag`
`Rigidbody.angularDrag`	`Timeline.rigidbody.angularDrag`
`Rigidbody.isKinematic`	`Timeline.rigidbody.isKinematic`
`Rigidbody.useGravity` `Rigidbody2D.gravityScale`	`Timeline.rigidbody.useGravity` `Timeline.rigidbody2D.gravityScale`

`Rigidbody.AddForce()`
`Rigidbody.AddRelativeForce()`
`Rigidbody.AddForceAtPosition()`
`Rigidbody.AddExplosionForce()`
`Rigidbody.AddTorque()`
`Rigidbody.AddRelativeTorque()`
	`Timeline.rigidbody.AddForce()`
	`Timeline.rigidbody.AddRelativeForce()`
	`Timeline.rigidbody.AddForceAtPosition()`
	`Timeline.rigidbody.AddExplosionForce()`
	`Timeline.rigidbody.AddTorque()`
	`Timeline.rigidbody.AddRelativeTorque()`

Wind Zones

Instead of ...	Use...
`WindZone.windMain`	`Timeline.windZone.windMain`
`WindZone.windTurbulence`	`Timeline.windZone.windTurbulence`
`WindZone.windPulseMagnitude`	`Timeline.windZone.windPulseMagnitude`
`WindZone.windPulseFrequency`	`Timeline.windZone.windPulseFrequency`

Did you spot any error in the migration tables?
If so, please report it in the forum!