

uNature

2.1

Generated by Doxygen 1.8.11

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	7
3.1	Class List	7
4	Namespace Documentation	13
4.1	uNature Namespace Reference	13
4.2	uNature.Core Namespace Reference	13
4.3	uNature.Core.ClassExtensions Namespace Reference	13
4.4	uNature.Core.Collections Namespace Reference	14
4.5	uNature.Core.Editor Namespace Reference	14
4.6	uNature.Core.Editor.Helpers Namespace Reference	14
4.7	uNature.Core.Extensions Namespace Reference	14
4.8	uNature.Core.FoliageClasses Namespace Reference	15
4.9	uNature.Core.Math Namespace Reference	16
4.10	uNature.Core.Networking Namespace Reference	16
4.11	uNature.Core.Pooling Namespace Reference	16
4.12	uNature.Core.Sectors Namespace Reference	17
4.13	uNature.Core.Seekers Namespace Reference	17
4.14	uNature.Core.Settings Namespace Reference	18
4.14.1	Enumeration Type Documentation	18
4.14.1.1	UNSettingCategories	18
4.15	uNature.Core.Targets Namespace Reference	18
4.16	uNature.Core.Terrains Namespace Reference	18
4.17	uNature.Core.Threading Namespace Reference	19
4.18	uNature.Core.Utility Namespace Reference	19
4.19	uNature.Demo Namespace Reference	20
4.20	uNature.Wrappers Namespace Reference	20
4.21	uNature.Wrappers.Linq Namespace Reference	20

5	Class Documentation	21
5.1	uNature.Core.FoliageClasses.BaseInteraction Class Reference	21
5.1.1	Member Function Documentation	22
5.1.1.1	OnDisable()	22
5.1.1.2	OnPositionChanged(Vector3 newPosition)	22
5.1.1.3	UpdateInteraction(FoliageReceiver receiver)	22
5.1.1.4	UpdateInteraction(FoliageReceiver receiver, Vector2 normalizedPosition)	22
5.2	uNature.Core.Utility.BasePrototypeltem Class Reference	23
5.2.1	Detailed Description	23
5.3	uNature.Core.Networking.BaseUNNetworkData Class Reference	23
5.3.1	Member Function Documentation	24
5.3.1.1	UnPack()	24
5.4	uNature.Core.Sectors.Chunk Class Reference	24
5.4.1	Detailed Description	26
5.4.2	Member Function Documentation	26
5.4.2.1	Awake()	26
5.4.2.2	Contains(Vector2 point, float offset)	26
5.4.2.3	Contains(Vector3 point, float offset)	26
5.4.2.4	CreateChunk< T >(Sector sector, Vector2 position, int x, int z, Vector2 scale, short chunkID)	26
5.4.2.5	OnCreated()	27
5.4.2.6	OnDisable()	27
5.4.2.7	OnDrawGizmos()	27
5.4.2.8	OnEnable()	27
5.4.2.9	OnSizeChanged()	27
5.4.2.10	ResetChunk()	28
5.5	uNature.Core.Sectors.ChunkObject Class Reference	28
5.6	uNature.Core.FoliageClasses.FoliageChunk Class Reference	28
5.6.1	Member Function Documentation	29
5.6.1.1	OnCreated()	29
5.6.1.2	OnDisable()	29

5.6.1.3	OnSizeChanged()	29
5.7	uNature.Core.FoliageClasses.FoliageCore_Chunk Class Reference	29
5.7.1	Member Function Documentation	30
5.7.1.1	Awake()	30
5.7.1.2	GetOrCreateFoliageManagerInstance()	30
5.7.1.3	InBounds(Vector3 normalizedPosition, float distance)	31
5.7.1.4	OnCreated()	31
5.7.1.5	OnDrawGizmos()	31
5.7.1.6	OnSizeChanged()	31
5.7.2	Property Documentation	31
5.7.2.1	isFoliageInstanceAttached	31
5.8	uNature.Core.FoliageClasses.FoliageCore_MainManager Class Reference	32
5.8.1	Member Function Documentation	33
5.8.1.1	Awake()	33
5.8.1.2	CheckChunkInBounds(int chunkID)	33
5.8.1.3	DestroyManager()	34
5.8.1.4	GetChunkID(float x, float z)	34
5.8.1.5	GetDetailLayer(int worldX, int worldZ, int sizeX, int sizeZ, int prototypeIndex)	34
5.8.1.6	InitializeAndCreateIfNotFound()	34
5.8.1.7	InsertFoliageFromTerrain(Terrain terrain, bool saveImmediately)	35
5.8.1.8	OnDisable()	35
5.8.1.9	RemoveGrassMap(FoliagePrototype prototype)	35
5.8.1.10	ResetGrassMap(List< FoliagePrototype > prototypes)	35
5.8.1.11	SaveDelayedMaps()	35
5.8.1.12	SetDetailLayer(int worldX, int worldZ, int sizeX, int sizeZ, int prototypeIndex, byte[,] densities)	35
5.8.1.13	Update()	36
5.8.1.14	UpdateHeights(int x, int z, int scaleX, int scaleZ)	36
5.8.1.15	UpdateHeightsOnTerrain(Terrain terrain)	36
5.8.1.16	WarmUpGrassMaps()	36
5.8.1.17	WarmUpGrassMaps(FoliageCore_Chunk[] specificChunks)	37

5.9	uNature.Core.FoliageClasses.FoliageCore_Sector Class Reference	37
5.9.1	Member Function Documentation	37
5.9.1.1	OnChunkCreated(Chunk chunk)	37
5.9.1.2	OnResolutionChanged()	38
5.9.1.3	OnStartCreatingChunks()	38
5.10	uNature.Core.FoliageClasses.FoliageDB Class Reference	38
5.10.1	Detailed Description	39
5.10.2	Member Function Documentation	39
5.10.2.1	AddPrototype(Texture2D texture, GameObject prefab, float minWidth, float minHeight, float maxWidth, float maxHeight, float spread, int layer, Color healthyColor, Color dryColor)	39
5.10.2.2	AddPrototype(DetailPrototype detailPrototype)	39
5.10.2.3	AddPrototype(Texture2D texture)	39
5.10.2.4	AddPrototype(GameObject prefab)	39
5.10.2.5	RemovePrototype(FoliagePrototype prototype)	39
5.10.2.6	UpdateShaderGeneralSettings()	39
5.10.2.7	UpdateShaderWindSettings()	40
5.10.3	Property Documentation	40
5.10.3.1	instance	40
5.11	uNature.Core.FoliageClasses.FoliageDynamicSurface Class Reference	40
5.12	uNature.Core.Utility.FoliageGrassMap Class Reference	41
5.12.1	Detailed Description	42
5.12.2	Member Function Documentation	42
5.12.2.1	Dispose()	42
5.12.2.2	GetDensity(int x, int z)	42
5.12.2.3	MarkDensitiesDirty()	42
5.12.2.4	SafeGetMap(FoliagePrototype prototype)	42
5.12.2.5	SetDensity(int x, int z, byte density)	43
5.12.2.6	SetDensityFast(int index, byte density)	43
5.12.2.7	UpdateGrassMaps(FoliageManagerInstance mInstance)	43
5.13	uNature.Core.Utility.FoliageHeightMap Class Reference	43

5.13.1 Detailed Description	44
5.14 uNature.Core.FoliageClasses.FoliageLODLevel Struct Reference	44
5.15 uNature.Core.FoliageClasses.FoliageManagerInstance Class Reference	44
5.15.1 Member Function Documentation	46
5.15.1.1 ForceMapsRestore()	46
5.15.1.2 InverseCord(float x, float addOffset)	46
5.15.1.3 InverseCordCustom(float x, float addOffset, float multiplier)	46
5.15.1.4 InverseCordCustomFloat(float x, float addOffset, float multiplier)	46
5.15.1.5 InverseCordFloat(float x, float addOffset)	47
5.15.1.6 OnDisable()	47
5.15.1.7 TransformCord(float x, float removeOffset)	47
5.15.1.8 TransformCordCustom(float x, float removeOffset, float multiplier)	47
5.15.1.9 TransformCordCustomFloat(float x, float removeOffset, float multiplier)	48
5.15.1.10 TransformCordFloat(float x, float removeOffset)	48
5.16 uNature.Core.FoliageClasses.FoliageMesh Class Reference	48
5.17 uNature.Core.FoliageClasses.FoliageMeshInstance Class Reference	49
5.17.1 Member Function Documentation	50
5.17.1.1 CreateFoliageInstances(int prototypeIndex, List< byte > densities, out Foliage↔ MeshInstancesGroup meshGroup, FoliageResolutions resolution)	50
5.17.1.2 GetDensity(float positionX, float positionZ, byte maxDensity, Color32[] mapPixels, int mapWidth)	50
5.17.1.3 MarkDensitiesDirty()	50
5.18 uNature.Core.FoliageClasses.FoliageMeshInstancesGroup Class Reference	50
5.19 uNature.Core.FoliageClasses.FoliageMeshManager Class Reference	51
5.19.1 Member Function Documentation	52
5.19.1.1 CreateNewMesh()	52
5.19.1.2 DestroyMeshInstance(int prototypeID)	52
5.19.1.3 DestroyMeshInstances()	53
5.19.1.4 GenerateFoliageMeshInstanceForIndex(int prototypeIndex, FoliageResolutions resolution)	53
5.19.1.5 GenerateFoliageMeshInstances()	53
5.19.1.6 GenerateFoliageMeshInstances(FoliageResolutions resolution)	53

5.19.1.7	GenerateFoliageMeshInstances(int prototypeID)	53
5.19.1.8	OnDisable()	54
5.19.1.9	RegenerateQueueInstances()	54
5.19.1.10	Update()	54
5.19.1.11	UpdateMeshBounds(Vector3 centerPos)	54
5.20	uNature.Core.Utility.FoliageNormalMap Class Reference	54
5.20.1	Detailed Description	54
5.20.2	Member Function Documentation	55
5.20.2.1	Save()	55
5.21	uNature.Core.FoliageClasses.FoliagePrototype Class Reference	55
5.21.1	Member Function Documentation	57
5.21.1.1	ApplyColorMap(Texture2D map, Texture2D normalMap)	57
5.21.1.2	ApplyGrassMap(Texture2D map)	57
5.21.1.3	ApplyWind()	57
5.21.1.4	CreatePrototype(Texture2D texture, GameObject prefab, float minWidth, float minHeight, float maxWidth, float maxHeight, float spread, int layer, int id, Color healthyColor, Color dryColor)	57
5.21.1.5	GetPreview()	57
5.21.1.6	UpdateManagerInformation()	58
5.22	uNature.Core.FoliageClasses.FoliageReceiver Class Reference	58
5.22.1	Member Function Documentation	59
5.22.1.1	OnDisable()	59
5.22.1.2	Update()	59
5.23	uNature.Core.FoliageClasses.FoliageSector Class Reference	59
5.23.1	Detailed Description	60
5.23.2	Member Function Documentation	60
5.23.2.1	OnChunkCreated(Chunk chunk)	60
5.23.2.2	OnStartCreatingChunks()	60
5.24	uNature.Core.Utility.FoliageWorldMaps Class Reference	60
5.24.1	Member Function Documentation	61
5.24.1.1	ClampNegativeIntoPositive(float value)	61

5.24.1.2	GetHeight(int index)	62
5.24.1.3	NormalizeHeight(float worldHeight)	62
5.24.1.4	ReGenerateGlobally()	62
5.24.1.5	RestoreChanges()	62
5.24.1.6	Save()	62
5.24.1.7	SaveAllMaps()	63
5.24.1.8	SetPixels32Delayed()	63
5.24.1.9	TransformNormals(Vector3 normal)	63
5.24.1.10	UpdateHeightAndNormal(int index, float height)	63
5.24.1.11	UpdateHeightAndNormal(int index, Vector3 normal)	63
5.24.1.12	UpdateHeightAndNormal(int index, float height, Vector3 normal)	63
5.24.1.13	UpdateHeightsAndNormals(float x, float z, int sizeX, int sizeZ, bool save)	64
5.24.1.14	UpdateHeightsAndNormals(bool save)	64
5.25	uNature.Core.FoliageClasses.GPUMesh Class Reference	64
5.25.1	Detailed Description	64
5.25.2	Member Data Documentation	65
5.25.2.1	LODMeshInstances	65
5.26	uNature.Core.FoliageClasses.GPUMeshLOD Class Reference	65
5.26.1	Detailed Description	65
5.27	uNature.Core.Sectors.GrassLODLevel Class Reference	65
5.27.1	Detailed Description	66
5.28	uNature.Core.Pooling.HarvestableTIPoolItem Class Reference	66
5.28.1	Detailed Description	67
5.28.2	Member Function Documentation	68
5.28.2.1	Awake()	68
5.28.2.2	HandleColliderDeath()	68
5.28.2.3	HandleHealthChange(int damage)	68
5.28.2.4	HandleTreeInstanceDeath()	68
5.28.2.5	Hit()	68
5.28.2.6	Hit(int damage)	68

5.28.2.7	OnPool()	68
5.28.2.8	OnReturnedToPool()	69
5.28.3	Member Data Documentation	69
5.28.3.1	canHarvestCollider	69
5.28.3.2	maxHealth	69
5.28.3.3	minHealth	69
5.28.4	Event Documentation	69
5.28.4.1	OnItemDamagedEvent	69
5.28.4.2	OnItemPooledEvent	69
5.28.4.3	OnItemReturnedToPoolEvent	69
5.29	uNature.Core.Pooling.IHarvestableItem Interface Reference	70
5.30	uNature.Wrappers.Linq.IOrderedEnumerable< TElement > Interface Template Reference	70
5.31	uNature.Core.Pooling.IPoolComponent Interface Reference	71
5.32	uNature.Core.Threading.IThreadTask Interface Reference	71
5.32.1	Detailed Description	71
5.33	uNature.Core.IUTCPhysicsIgnored Interface Reference	71
5.33.1	Detailed Description	72
5.34	uNature.Core.Extensions.MethodHelperAttribute Class Reference	72
5.35	uNature.Wrappers.Linq.OrderedEnumerable< TElement > Class Template Reference	72
5.36	uNature.Wrappers.Linq.OrderedSequence< TElement, TKey > Class Template Reference	73
5.37	uNature.Core.FoliageClasses.PaintBrush Class Reference	73
5.38	uNature.Core.Pooling.Pool Class Reference	74
5.38.1	Detailed Description	75
5.38.2	Member Function Documentation	75
5.38.2.1	AddToPool(PoolItem item, int itemID, int itemID_Offset)	75
5.38.2.2	CreatePool(string name, GameObject requester)	76
5.38.2.3	GetPoolOfItem(int itemUID, int itemID_Offset)	76
5.38.2.4	IsAlreadyPooled(int uid)	76
5.38.2.5	PoolItem(PoolItem PoolItem, bool locked, int uid)	76
5.38.2.6	RemoveDuplications(string name)	77

5.38.2.7	RemoveFromPool(PoolItem item)	77
5.38.2.8	ResetFarAway()	77
5.38.2.9	ReturnToPool(PoolItem item, bool force)	77
5.38.2.10	TryGetType< T >()	77
5.38.2.11	TryPool< T >(int itemUID, int itemID_Offset, int uid, bool locked)	78
5.38.2.12	TryResetOnUID(int uid, bool forceReset)	78
5.38.3	Member Data Documentation	78
5.38.3.1	items	78
5.38.3.2	owner	79
5.39	uNature.Core.Pooling.PoolItem Class Reference	79
5.39.1	Detailed Description	80
5.39.2	Member Function Documentation	80
5.39.2.1	Awake()	80
5.39.2.2	MoveItem(Vector3 position)	80
5.39.2.3	OnCreated()	81
5.39.2.4	OnDisable()	81
5.39.2.5	OnEnable()	81
5.39.2.6	OnPool()	81
5.39.2.7	OnReturnedToPool()	81
5.39.3	Member Data Documentation	81
5.39.3.1	_gameObject	81
5.39.3.2	itemID_Offset	81
5.39.3.3	locked	82
5.39.3.4	Pool	82
5.39.3.5	realItemID	82
5.39.3.6	uid	82
5.39.3.7	used	82
5.39.4	Property Documentation	82
5.39.4.1	itemID	82
5.39.4.2	PoolTypes	82

5.40	uNature.Wrappers.Linq.QuickSort< TElement > Class Template Reference	82
5.41	uNature.Core.FoliageClasses.RenderingQueue Class Reference	83
5.41.1	Detailed Description	83
5.42	uNature.Core.FoliageClasses.RenderingQueueInstance Class Reference	83
5.42.1	Detailed Description	84
5.43	uNature.Core.FoliageClasses.RenderingQueueMeshInstanceSimulator Struct Reference	84
5.43.1	Detailed Description	84
5.44	uNature.Core.FoliageClasses.RenderingQueueReceiver Class Reference	85
5.44.1	Detailed Description	85
5.45	uNature.Core.Sectors.Sector Class Reference	85
5.45.1	Detailed Description	87
5.45.2	Member Function Documentation	87
5.45.2.1	ApplicationQuit()	87
5.45.2.2	Awake()	87
5.45.2.3	GenerateSector< T, T1 >(Transform owner, Vector3 bounds, T sector, int res)	87
5.45.2.4	getChunk(Vector2 pos, float offset)	87
5.45.2.5	getChunk(Vector3 pos, float offset)	88
5.45.2.6	getChunk(Vector3 pos)	88
5.45.2.7	getChunks(Vector2 pos, float offset, bool sortResult)	88
5.45.2.8	getChunks(Vector3 pos, float offset, bool sortResult)	89
5.45.2.9	OnChunkCreated(Chunk chunk)	89
5.45.2.10	OnCreated(Transform owner, int resolution)	89
5.45.2.11	OnResolutionChanged()	89
5.45.2.12	OnStartCreatingChunks()	90
5.45.2.13	ResetChunks()	90
5.46	uNature.Wrappers.Linq.SortContext< TElement > Class Template Reference	90
5.47	uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey > Class Template Reference	91
5.48	uNature.Core.Pooling.TerrainPoolItem Class Reference	91
5.48.1	Detailed Description	92
5.48.2	Member Function Documentation	92

5.48.2.1	ConvertTreeInstanceOnTerrain(Terrain terrain, int treeInstanceUID)	92
5.48.2.2	MoveItem(Vector3 position)	92
5.48.2.3	RemoveTreeInstanceFromTerrain(Terrain terrain, int treeInstanceUID)	93
5.48.2.4	RestoreTreeInstanceToTerrain(Terrain terrain, int treeInstanceUID)	93
5.48.3	Member Data Documentation	93
5.48.3.1	_terrain	93
5.48.3.2	canModify	93
5.48.3.3	canRestore	93
5.48.3.4	isCollider	93
5.49	uNature.Core.Threading.ThreadItem Class Reference	94
5.49.1	Detailed Description	94
5.49.2	Member Function Documentation	95
5.49.2.1	OnDisable()	95
5.49.2.2	OnPositionChanged(Vector3 newPosition)	95
5.49.2.3	Update()	95
5.49.2.4	UpdateItem()	95
5.49.3	Member Data Documentation	95
5.49.3.1	_threadItems	95
5.50	uNature.Core.Threading.ThreadTask Class Reference	95
5.50.1	Detailed Description	96
5.51	uNature.Core.Threading.ThreadTask Class Reference	96
5.51.1	Detailed Description	96
5.52	uNature.Core.Threading.ThreadTask Class Reference	96
5.52.1	Detailed Description	96
5.53	uNature.Core.Threading.ThreadTask Class Reference	96
5.53.1	Detailed Description	96
5.54	uNature.Core.Threading.ThreadTask Class Reference	97
5.54.1	Detailed Description	97
5.55	uNature.Core.Sectors.TIChunk Class Reference	97
5.55.1	Member Function Documentation	98

5.55.1.1	AddTreeInstance(int instanceID, Vector3 terrainSize, TreeInstance treeInstance, TerrainData terrainData, Vector3 terrainPos, UNTerrainSector sector)	98
5.55.1.2	Awake()	98
5.55.1.3	CheckForNearbyTreeInstances(UNSeeker seeker, UNTerrain terrain)	98
5.55.1.4	GenerateTreeInstances(TreeInstance[] trees, Vector3 terrainSize, TerrainData t↔Data, Vector3 terrainPos)	98
5.55.1.5	OnCreated()	99
5.55.1.6	OnDrawGizmos()	99
5.55.1.7	OnSizeChanged()	99
5.55.1.8	ResetChunk()	99
5.56	uNature.Core.FoliageClasses.TouchBending Class Reference	99
5.56.1	Member Function Documentation	100
5.56.1.1	OnDisable()	100
5.56.1.2	OnPositionChanged(Vector3 newPosition)	100
5.57	uNature.Core.FoliageClasses.TouchBendingEditor Class Reference	101
5.58	uNature.Core.Sectors.TreeFetchingTask_MultiThreaded Struct Reference	101
5.59	uNature.Demo.UN_FirstPersonController Class Reference	101
5.60	uNature.Core.Extensions.UN_ForgeNetworking Class Reference	102
5.61	uNature.Core.Extensions.UN_GAIA Class Reference	102
5.62	uNature.Core.Extensions.UN_GENA Class Reference	103
5.63	uNature.Core.Extensions.UN_MapMagic Class Reference	104
5.64	uNature.Demo.UN_MouseLook Class Reference	104
5.65	uNature.Core.Extensions.UN_PhotonBolt Class Reference	105
5.66	uNature.Core.Extensions.UN_PhotonCloud Class Reference	105
5.67	uNature.Core.Extensions.UN_TerrainComposer Class Reference	106
5.68	uNature.Core.Utility.UN_TerrainTexturePrototype Class Reference	106
5.68.1	Detailed Description	107
5.68.2	Member Function Documentation	107
5.68.2.1	GetPreview()	107
5.69	uNature.Core.Extensions.UN_UFPS Class Reference	107
5.70	uNature.Core.Extensions.UN_UNet Class Reference	108

5.71	uNature.Core.Extensions.UN_WorldStreamer Class Reference	109
5.72	uNature.Core.Utility.UNBrushUtility Class Reference	109
5.72.1	Detailed Description	110
5.72.2	Member Function Documentation	110
5.72.2.1	CheckSplatPaint(RaycastHit hit, Vector3 worldPosition, List< UN_TerrainTexturePrototype > chosenSplats)	110
5.72.2.2	DrawBrush(Texture2D brushTexture, Color brushColor, Vector3 originPosition, Quaternion originRotation, float brushSize)	110
5.72.2.3	Resize(Texture2D source, int newWidth, int newHeight)	111
5.73	uNature.Core.Utility.UNCombineInstance Struct Reference	111
5.74	uNature.Core.Utility.UNDictionary< T, T1 > Class Template Reference	111
5.75	uNature.Core.Collections.UNDimensionalList< T > Class Template Reference	112
5.75.1	Detailed Description	112
5.75.2	Member Function Documentation	112
5.75.2.1	ContainsKey(int key)	112
5.75.2.2	ContainsValue(T value)	112
5.75.2.3	TryAddKey(List< T > value)	113
5.75.3	Property Documentation	113
5.75.3.1	Count	113
5.75.3.2	this[int index]	113
5.76	uNature.Core.Editor.Helpers.UNEditorHelpers Class Reference	113
5.77	uNature.Core.Extensions.UNExtension Class Reference	114
5.77.1	Detailed Description	115
5.77.2	Member Function Documentation	115
5.77.2.1	GetLogo(UNExtension instance)	115
5.77.2.2	LoadMethods(UNExtension instance, Type type)	116
5.77.2.3	OpenAssetStore(UNExtension instance)	116
5.77.2.4	OpenDocs(UNExtension instance)	116
5.77.3	Member Data Documentation	116
5.77.3.1	HelperMethods	116
5.77.3.2	isViewed	116

5.77.4	Property Documentation	116
5.77.4.1	AssetDescription	116
5.77.4.2	AssetDocumentationName	117
5.77.4.3	AssetLogoName	117
5.77.4.4	AssetName	117
5.77.4.5	AssetNameSpace	117
5.77.4.6	AssetStoreAdress	117
5.77.4.7	Featured	117
5.77.4.8	isActivated	117
5.77.4.9	IsDefault	117
5.77.4.10	PublisherName	117
5.78	uNature.Core.Extensions.UNExtensionsEditor Class Reference	118
5.79	uNature.Core.Utility.UNFastList< T > Class Template Reference	118
5.79.1	Detailed Description	118
5.80	uNature.Core.FoliageClasses.UNFoliageEditor Class Reference	119
5.81	uNature.Core.FoliageClasses.UNFoliageManagerEditor Class Reference	119
5.82	uNature.Core.Collections.UNList< T > Class Template Reference	120
5.82.1	Detailed Description	120
5.82.2	Member Function Documentation	120
5.82.2.1	Add(T item)	120
5.82.2.2	Contains(System.Object item)	121
5.82.2.3	Remove(T item)	121
5.82.2.4	TryGet(System.Object similarItem)	121
5.82.3	Property Documentation	121
5.82.3.1	Count	121
5.82.3.2	this[int index]	121
5.83	uNature.Core.Utility.UNMap Class Reference	122
5.83.1	Detailed Description	123
5.84	uNature.Core.FoliageClasses.UNMeshData Class Reference	123
5.85	uNature.Core.Networking.UNNetworkData< T > Class Template Reference	123

5.85.1 Detailed Description	124
5.85.2 Member Function Documentation	125
5.85.2.1 Deserialize(byte[] bytes)	125
5.85.2.2 Equals(object obj)	125
5.85.2.3 GetHashCode()	125
5.85.2.4 Pack< T1, T2 >(Terrain terrain, int treeInstanceId, int health, PacketType type)	125
5.85.2.5 SendToClients()	126
5.85.2.6 SendToConnection(T connection)	126
5.85.2.7 SendToOthers()	127
5.85.2.8 SendToServer()	127
5.85.2.9 Serialize()	127
5.85.2.10 UnPack()	127
5.86 uNature.Core.Networking.UNNetworkManager< T1, T2 > Class Template Reference	127
5.86.1 Detailed Description	128
5.86.2 Constructor & Destructor Documentation	129
5.86.2.1 UNNetworkManager(MonoBehaviour managerInstance)	129
5.86.3 Member Function Documentation	129
5.86.3.1 Awake()	129
5.86.3.2 CheckForStreamingBufferedUpdates()	129
5.86.3.3 OnClientConnected(T1 conn)	129
5.86.3.4 OnHarvestableTreeInstancePooled(HarvestableTIPoolItem instance)	129
5.86.3.5 OnItemDamaged(HarvestableTIPoolItem item, int damage)	130
5.86.3.6 SendEvent(UNNetworkData< T1 > instance)	130
5.86.3.7 SendToClients(UNNetworkData< T1 > instance)	130
5.86.3.8 SendToConnection(T1 connection, UNNetworkData< T1 > instance)	130
5.86.3.9 SendToOthers(UNNetworkData< T1 > instance)	131
5.86.3.10 SendToServer(UNNetworkData< T1 > instance)	131
5.86.3.11 UpdatePermissions()	131
5.86.4 Member Data Documentation	131
5.86.4.1 manager	131

5.86.5	Property Documentation	131
5.86.5.1	bufferedData	131
5.86.5.2	isAuth	131
5.86.5.3	isServer	132
5.87	uNature.Core.Networking.UNNetworkPlayerController Class Reference	132
5.88	uNature.Core.UNPhysicsHit_Grass Struct Reference	132
5.88.1	Detailed Description	133
5.89	uNature.Core.UNPhysicsHitsArray Class Reference	133
5.89.1	Detailed Description	133
5.90	uNature.Core.UNPhysicsObject Struct Reference	133
5.90.1	Detailed Description	134
5.90.2	Member Function Documentation	134
5.90.2.1	DrawShape(Matrix4x4 matrix)	134
5.90.2.2	OnDrawGizmos()	134
5.90.2.3	Raycast(Ray ray, out UNPhysicsHit_Grass _hit, LayerMask mask)	134
5.90.2.4	UpdateBounds()	135
5.91	uNature.Core.UNPhysicsTemplate Struct Reference	135
5.92	uNature.Core.Seekers.UNSeeker Class Reference	136
5.92.1	Detailed Description	137
5.92.2	Member Function Documentation	137
5.92.2.1	CheckTargetsOnMove()	137
5.92.2.2	HarvestChecks()	137
5.92.2.3	Start()	137
5.92.2.4	Update()	137
5.92.3	Member Data Documentation	137
5.92.3.1	_treesCheckDistance	137
5.92.3.2	attackTrees	137
5.92.3.3	lastMovement	138
5.92.3.4	raycastDistance	138
5.92.3.5	raycastMask	138

5.92.3.6	seekingDistance	138
5.93	uNature.Core.Seekers.UNSeekerEditor Class Reference	138
5.94	uNature.Core.Settings.UNSetting Class Reference	138
5.94.1	Detailed Description	139
5.94.2	Member Function Documentation	139
5.94.2.1	DrawGUI()	139
5.95	uNature.Core.Settings.UNSettingAttribute Class Reference	139
5.95.1	Detailed Description	140
5.96	uNature.Core.Settings.UNSettingCategory Class Reference	140
5.96.1	Detailed Description	140
5.97	uNature.Core.Settings.UNSettings Class Reference	141
5.97.1	Detailed Description	141
5.97.2	Member Function Documentation	141
5.97.2.1	Log(string context)	141
5.97.2.2	ResetDefaults()	142
5.97.3	Member Data Documentation	142
5.97.3.1	fileName	142
5.97.3.2	ProjectName	142
5.97.3.3	ProjectVersion	142
5.97.4	Property Documentation	142
5.97.4.1	ProjectPath	142
5.98	uNature.Core.Settings.UNSettingsEditor Class Reference	142
5.99	uNature.Core.Targets.UNTarget Class Reference	143
5.99.1	Detailed Description	144
5.99.2	Member Function Documentation	145
5.99.2.1	Awake()	145
5.99.2.2	Check(Seekers.UNSeeker seeker, Vector3 seekerPos, float seekingDistance, bool isPlaying)	145
5.99.2.3	CheckTargets(UNSeeker seeker, float distance)	145
5.99.2.4	CreatePool(System.Type PoolItemType)	145
5.99.2.5	FixPosition(Vector3 position)	145

5.99.2.6	InDistance(UNSeeker seeker)	146
5.99.2.7	OnDisable()	146
5.99.2.8	OnDrawGizmos()	146
5.99.2.9	OnEnable()	146
5.99.2.10	Update()	146
5.99.3	Member Data Documentation	147
5.99.3.1	Pool	147
5.99.3.2	PoolAmount	147
5.99.3.3	PoolTypeSerializedName	147
5.99.3.4	worldTargets	147
5.99.4	Property Documentation	147
5.99.4.1	useMultithreadedCheck	147
5.100	uNature.Core.Targets.UNTargetEditor Class Reference	147
5.101	uNature.Core.Terrains.UNTerrain Class Reference	148
5.101.1	Detailed Description	149
5.101.2	Member Function Documentation	149
5.101.2.1	Awake()	149
5.101.2.2	Check(Seekers.UNSeeker seeker, Vector3 seekerPos, float seekingDistance, bool isPlaying)	149
5.101.2.3	CheckForTreeInstancesRespawns()	150
5.101.2.4	CreatePool(System.Type PoolItemType)	150
5.101.2.5	FixPosition(Vector3 position)	150
5.101.2.6	GenerateSector(int sectorResolution, bool multiThread)	150
5.101.2.7	GenerateSector(int sectorResolution)	150
5.101.2.8	InDistance(Seekers.UNSeeker seeker)	151
5.101.2.9	OnDisable()	151
5.101.2.10	OnEnable()	151
5.101.2.11	OnPositionChanged(Vector3 newPosition)	151
5.101.2.12	OnTerrainChanged(int changedFlags)	151
5.101.3	Member Data Documentation	152
5.101.3.1	collidersPoolItemInstanceIncrease	152

5.101.3.2 distanceOffset	152
5.101.3.3 removedTreeInstanceHeight	152
5.101.3.4 terrains	152
5.101.3.5 verifyTreeInstancesChangeRoutine	152
5.101.4 Property Documentation	152
5.101.4.1 manageGrass	152
5.101.4.2 manageTrees	152
5.101.4.3 sectorResolution	153
5.102uNature.Core.Terrains.UNTerrainData Class Reference	153
5.102.1 Detailed Description	154
5.102.2 Member Function Documentation	154
5.102.2.1 ApplyBackup(Terrain terrain)	154
5.102.2.2 Backup()	154
5.102.2.3 CheckForTreePrototypesChange()	154
5.102.2.4 DeleteBackup()	154
5.102.2.5 GetInstance(TerrainData terrainData)	154
5.102.2.6 GetPrototype(TreePrototype prototype)	155
5.102.2.7 Initialize()	155
5.102.2.8 SendUpdateEventToLinkedTerrains(TerrainChangedFlags flag)	155
5.102.2.9 UpdateMultithreadedVariables()	155
5.102.3 Property Documentation	155
5.102.3.1 heights	155
5.102.3.2 terrainData	155
5.103uNature.Core.Terrains.UNTerrainEditor Class Reference	156
5.104uNature.Core.Sectors.UNTerrainSector Class Reference	156
5.104.1 Detailed Description	157
5.104.2 Member Function Documentation	157
5.104.2.1 ApplicationQuit()	157
5.104.2.2 Awake()	158
5.104.2.3 FetchTreeInstances(bool useUNThread, System.Action OnFinish)	158

5.104.2.4 OnChunkCreated(Chunk chunk)	158
5.104.2.5 OnCreated(Transform owner, int resolution)	158
5.104.2.6 OnStartCreatingChunks()	158
5.105uNature.Core.Threading.UNThreadManager Class Reference	159
5.105.1 Detailed Description	160
5.105.2 Member Function Documentation	160
5.105.2.1 DelayActionFrames(int frames, IThreadTask task)	160
5.105.2.2 DelayActionSeconds(IThreadTask task, float time)	160
5.105.2.3 OnThreadProcess(System.Object processObject)	160
5.105.2.4 RunOnThread(IThreadTask action)	160
5.105.2.5 RunOnUnityThread(IThreadTask action)	160
5.105.2.6 UpdateThreadItems()	161
5.105.3 Member Data Documentation	161
5.105.3.1 updateThreadItemsTime	161
5.105.4 Property Documentation	161
5.105.4.1 threadEnabled	161
5.105.4.2 threadWorkersCount	161
5.106uNature.Core.Terrains.UNTreePrototype Class Reference	161
5.106.1 Detailed Description	162
5.106.2 Constructor & Destructor Documentation	162
5.106.2.1 UNTreePrototype(TreePrototype prototype)	162
5.106.3 Member Function Documentation	163
5.106.3.1 CheckForMissings(List< UNTreePrototype > items, TreePrototype[] prototypes)	163
5.106.3.2 Equals(object obj)	163
5.106.3.3 GetHashCode()	163
5.106.3.4 GetPreview()	163
5.106.4 Member Data Documentation	164
5.106.4.1 isMissing	164
5.106.4.2 prototypeObject	164
5.106.5 Property Documentation	164
5.106.5.1 isEnabled	164
5.107uNature.Core.Vector2i Struct Reference	164
5.108uNature.Core.Utility.Vector3_XZ_FAST Struct Reference	165
5.109uNature.Core.FoliageClasses.WindSettings Class Reference	165
5.110uNature.Core.FoliageClasses.WindZones Class Reference	165

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

uNature	13
uNature.Core	13
uNature.Core.ClassExtensions	13
uNature.Core.Collections	14
uNature.Core.Editor	14
uNature.Core.Editor.Helpers	14
uNature.Core.Extensions	14
uNature.Core.FoliageClasses	15
uNature.Core.Math	16
uNature.Core.Networking	16
uNature.Core.Pooling	16
uNature.Core.Sectors	17
uNature.Core.Seekers	17
uNature.Core.Settings	18
uNature.Core.Targets	18
uNature.Core.Terrains	18
uNature.Core.Threading	19
uNature.Core.Utility	19
uNature.Demo	20
uNature.Wrappers	20
uNature.Wrappers.Linq	20

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Attribute	
uNature.Core.Extensions.MethodHelperAttribute	72
uNature.Core.Settings.UNSettingAttribute	139
uNature.Core.Utility.BasePrototypeItem	23
uNature.Core.FoliageClasses.FoliagePrototype	55
uNature.Core.FoliageClasses.PaintBrush	73
uNature.Core.Terrains.UNTreePrototype	161
uNature.Core.Utility.UN_TerrainTexturePrototype	106
uNature.Core.Networking.BaseUNNetworkData	23
uNature.Core.Networking.UNNetworkData< T >	123
uNature.Core.Sectors.ChunkObject	28
Editor	
uNature.Core.FoliageClasses.TouchBendingEditor	101
uNature.Core.FoliageClasses.UNFoliageManagerEditor	119
uNature.Core.Seekers.UNSeekerEditor	138
uNature.Core.Targets.UNTargetEditor	147
uNature.Core.Terrains.UNTerrainEditor	156
EditorWindow	
uNature.Core.Extensions.UNExtensionsEditor	118
uNature.Core.FoliageClasses.UNFoliageEditor	119
uNature.Core.Settings.UNSettingsEditor	142
uNature.Core.FoliageClasses.FoliageLODLevel	44
uNature.Core.FoliageClasses.FoliageMesh	48
uNature.Core.FoliageClasses.FoliageMeshInstance	49
uNature.Core.FoliageClasses.FoliageMeshInstancesGroup	50
uNature.Core.Utility.FoliageWorldMaps	60
uNature.Core.FoliageClasses.GPUMesh	64
uNature.Core.FoliageClasses.GPUMeshLOD	65
uNature.Core.Sectors.GrassLODLevel	65
IComparer	
uNature.Wrappers.Linq.SortContext< TElement >	90
uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey >	91
IEnumerable	
uNature.Wrappers.Linq.IOrderedEnumerable< TElement >	70

uNature.Wrappers.Linq.OrderedEnumerable< TElement >	72
uNature.Wrappers.Linq.OrderedSequence< TElement, TKey >	73
uNature.Core.Pooling.IHarvestableItem	70
uNature.Core.Pooling.HarvestableTIPoolItem	66
uNature.Core.Pooling.IPoolComponent	71
uNature.Core.Threading.IThreadTask	71
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >	97
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >	97
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >	97
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >	97
uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >	97
uNature.Core.IUTCPPhysicsIgnored	71
MonoBehaviour	
uNature.Core.FoliageClasses.FoliageDynamicSurface	40
uNature.Core.Networking.UNNetworkPlayerController	132
uNature.Core.Pooling.Pool	74
uNature.Core.Sectors.Chunk	24
uNature.Core.FoliageClasses.FoliageChunk	28
uNature.Core.FoliageClasses.FoliageCore_Chunk	29
uNature.Core.Sectors.TIChunk	97
uNature.Core.Sectors.Sector	85
uNature.Core.FoliageClasses.FoliageCore_Sector	37
uNature.Core.FoliageClasses.FoliageSector	59
uNature.Core.Sectors.UNTerrainSector	156
uNature.Core.Threading.ThreadItem	94
uNature.Core.FoliageClasses.BaseInteraction	21
uNature.Core.FoliageClasses.TouchBending	99
uNature.Core.FoliageClasses.WindZones	165
uNature.Core.FoliageClasses.FoliageManagerInstance	44
uNature.Core.FoliageClasses.FoliageReceiver	58
uNature.Core.Seekers.UNSeeker	136
uNature.Core.Pooling.PoolItem	79
uNature.Core.Pooling.TerrainPoolItem	91
uNature.Core.Pooling.HarvestableTIPoolItem	66
uNature.Core.Targets.UNTarget	143
uNature.Core.FoliageClasses.FoliageMeshManager	51
uNature.Core.FoliageClasses.FoliageCore_MainManager	32
uNature.Core.Terrains.UNTerrain	148
uNature.Core.Threading.UNThreadManager	159
uNature.Core.Utility.UNBrushUtility	109
uNature.Demo.UN_FirstPersonController	101
uNature.Wrappers.Linq.QuickSort< TElement >	82
uNature.Core.FoliageClasses.RenderingQueue	83
uNature.Core.FoliageClasses.RenderingQueueInstance	83
uNature.Core.FoliageClasses.RenderingQueueMeshInstanceSimulator	84
uNature.Core.FoliageClasses.RenderingQueueReceiver	85
ScriptableObject	
uNature.Core.FoliageClasses.FoliageDB	38
uNature.Core.Settings.UNSettings	141
uNature.Core.Terrains.UNTerrainData	153
uNature.Core.Threading.ThreadTask	97
uNature.Core.Threading.ThreadTask< T >	97
uNature.Core.Threading.ThreadTask< T, T1 >	97
uNature.Core.Threading.ThreadTask< T, T1, T2 >	97
uNature.Core.Sectors.TreeFetchingTask_MultiThreaded	101
uNature.Demo.UN_MouseLook	104
uNature.Core.Utility.UNCombineInstance	111

uNature.Core.Utility.UNDictionary< T, T1 >	111
uNature.Core.Collections.UNDimensionalList< T >	112
uNature.Core.Collections.UNDimensionalList< int >	112
uNature.Core.Editor.Helpers.UNEditorHelpers	113
uNature.Core.Extensions.UNExtension	114
uNature.Core.Extensions.UN_ForgeNetworking	102
uNature.Core.Extensions.UN_GAIA	102
uNature.Core.Extensions.UN_GENA	103
uNature.Core.Extensions.UN_MapMagic	104
uNature.Core.Extensions.UN_PhotonBolt	105
uNature.Core.Extensions.UN_PhotonCloud	105
uNature.Core.Extensions.UN_TerrainComposer	106
uNature.Core.Extensions.UN_UFPS	107
uNature.Core.Extensions.UN_UNet	108
uNature.Core.Extensions.UN_WorldStreamer	109
uNature.Core.Utility.UNFastList< T >	118
uNature.Core.Utility.UNFastList< uNature.Core.FoliageClasses.RenderingQueueMeshInstanceSimulator >	118
uNature.Core.Collections.UNList< T >	120
uNature.Core.Collections.UNList< uNature.Core.Networking.BaseUNNetworkData >	120
uNature.Core.Utility.UNMap	122
uNature.Core.Utility.FoliageGrassMap	41
uNature.Core.Utility.FoliageHeightMap	43
uNature.Core.Utility.FoliageNormalMap	54
uNature.Core.FoliageClasses.UNMeshData	123
uNature.Core.Networking.UNNetworkManager< T1, T2 >	127
uNature.Core.UNPhysicsHit_Grass	132
uNature.Core.UNPhysicsHitsArray	133
uNature.Core.UNPhysicsObject	133
uNature.Core.UNPhysicsTemplate	135
uNature.Core.Settings.UNSetting	138
uNature.Core.Settings.UNSettingCategory	140
uNature.Core.Vector2i	164
uNature.Core.Utility.Vector3_XZ_FAST	165
uNature.Core.FoliageClasses.WindSettings	165

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

uNature.Core.FoliageClasses.BaseInteraction	21
uNature.Core.Utility.BasePrototypeItem	
An base prototype item which is used for the uNature ui utility.	23
uNature.Core.Networking.BaseUNNetworkData	23
uNature.Core.Sectors.Chunk	
part of the sector which contains information.	24
uNature.Core.Sectors.ChunkObject	28
uNature.Core.FoliageClasses.FoliageChunk	28
uNature.Core.FoliageClasses.FoliageCore_Chunk	29
uNature.Core.FoliageClasses.FoliageCore_MainManager	32
uNature.Core.FoliageClasses.FoliageCore_Sector	37
uNature.Core.FoliageClasses.FoliageDB	
The database class of the Foliage, holds a lot of important data such as Foliage prototypes, Foliage map and more.	38
uNature.Core.FoliageClasses.FoliageDynamicSurface	40
uNature.Core.Utility.FoliageGrassMap	
Channels:	41
uNature.Core.Utility.FoliageHeightMap	
Channels: R: Heights Channel #1 G: Heights Channel #2	43
uNature.Core.FoliageClasses.FoliageLODLevel	44
uNature.Core.FoliageClasses.FoliageManagerInstance	44
uNature.Core.FoliageClasses.FoliageMesh	48
uNature.Core.FoliageClasses.FoliageMeshInstance	49
uNature.Core.FoliageClasses.FoliageMeshInstancesGroup	50
uNature.Core.FoliageClasses.FoliageMeshManager	51
uNature.Core.Utility.FoliageNormalMap	
Channels: R: Heights Channel #1 G: Heights Channel #2	54
uNature.Core.FoliageClasses.FoliagePrototype	55
uNature.Core.FoliageClasses.FoliageReceiver	58
uNature.Core.FoliageClasses.FoliageSector	
An sector class dedicated only to Foliage.	59
uNature.Core.Utility.FoliageWorldMaps	60
uNature.Core.FoliageClasses.GPUMesh	
A class used to hold the gpu meshes	64
uNature.Core.FoliageClasses.GPUMeshLOD	
GPU Mesh Lods.	65

97

[uNature.Core.Threading.ThreadTask< T, T1, T2 >](#)

A thread task that takes 3 parameters.

Template Parameters

<i>T</i>	Type 1
<i>T1</i>	Type 2
<i>T2</i>	Type 3

97

[uNature.Core.Threading.ThreadTask< T, T1, T2, T3 >](#)

A thread task that takes 4 parameters.

Template Parameters

<i>T</i>	Type 1
<i>T1</i>	Type 2
<i>T2</i>	Type 3
<i>T3</i>	Type 4

97

[uNature.Core.Sectors.TIChunk](#) 97[uNature.Core.FoliageClasses.TouchBending](#) 99[uNature.Core.FoliageClasses.TouchBendingEditor](#) 101[uNature.Core.Sectors.TreeFetchingTask_MultiThreaded](#) 101[uNature.Demo.UN_FirstPersonController](#) 101[uNature.Core.Extensions.UN_ForgeNetworking](#) 102[uNature.Core.Extensions.UN_GAIA](#) 102[uNature.Core.Extensions.UN_GENA](#) 103[uNature.Core.Extensions.UN_MapMagic](#) 104[uNature.Demo.UN_MouseLook](#) 104[uNature.Core.Extensions.UN_PhotonBolt](#) 105[uNature.Core.Extensions.UN_PhotonCloud](#) 105[uNature.Core.Extensions.UN_TerrainComposer](#) 106[uNature.Core.Utility.UN_TerrainTexturePrototype](#)

The prototype of the terrain texture. [Used for listing the paintable surfaces] 106

[uNature.Core.Extensions.UN_UFPS](#) 107[uNature.Core.Extensions.UN_UNet](#) 108[uNature.Core.Extensions.UN_WorldStreamer](#) 109[uNature.Core.Utility.UNBrushUtility](#)

Using this class you can paint a brush on the scene. 109

[uNature.Core.Utility.UNCombineInstance](#) 111[uNature.Core.Utility.UNDictionary< T, T1 >](#) 111[uNature.Core.Collections.UNDimensionalList< T >](#)A 2 dimensional list which is used by certain mechanics in [uNature](#). 112[uNature.Core.Editor.Helpers.UNEditorHelpers](#) 113[uNature.Core.Extensions.UNExtension](#)

A uConstruct extension that will allow other 3d party systems to work with uConstruct. 114

[uNature.Core.Extensions.UNExtensionsEditor](#) 118[uNature.Core.Utility.UNFastList< T >](#)

A list that requires you to set 118

[uNature.Core.FoliageClasses.UNFoliageEditor](#) 119[uNature.Core.FoliageClasses.UNFoliageManagerEditor](#) 119[uNature.Core.Collections.UNList< T >](#)

A custom list which is used on some important interfaces in UN. 120

uNature.Core.Utility.UNMap	
The abstract Map class.	122
uNature.Core.FoliageClasses.UNMeshData	123
uNature.Core.Networking.UNNetworkData< T >	
A class which can be used for an abstract networking data.	

Template Parameters

<i>T</i>	The network connection type which the networking library uses.
----------	--

123

uNature.Core.Networking.UNNetworkManager< T1, T2 >	
A template for networking, which can be used by networking extensions to easily get the network- ing actions done.	

Template Parameters

<i>T1</i>	the targeted networking connection
<i>T2</i>	the type of the data

uNature.Core.Networking.UNNetworkPlayerController	127
uNature.Core.UNPhysicsHit_Grass	132
A class that holds the data for the hit data	132
uNature.Core.UNPhysicsHitsArray	
An custom array that holds all ray results in an array	133
uNature.Core.UNPhysicsObject	
This is a base class for a UCPysicsObject. Every class that inherits this class will be counted in the physics system.	133
uNature.Core.UNPhysicsTemplate	135
uNature.Core.Seekers.UNSeeker	
<i>Seekers</i> are basically GameObjects in the scene which should interact with the objects in the game.	136
uNature.Core.Seekers.UNSeekerEditor	138
uNature.Core.Settings.UNSetting	
A class which should be used on custom classes that needs to be shown and serialized as a setting.	138
uNature.Core.Settings.UNSettingAttribute	
The attribute of each setting which handles the drawing of the setting (generically).	139
uNature.Core.Settings.UNSettingCategory	
The class of a category which handles keeping hold of all of the categories and makes all of the reflection needed.	140
uNature.Core.Settings.UNSettings	
A class which handles certain settings of aspects in UN	141
uNature.Core.Settings.UNSettingsEditor	142
uNature.Core.Targets.UNTarget	
A target is what will be taken into account with the system. For example terrains.	143
uNature.Core.Targets.UNTargetEditor	147
uNature.Core.Terrains.UNTerrain	
A class that needs to be on each terrain that needs to be taken into account when managing the system.	148
uNature.Core.Terrains.UNTerrainData	
The terrain data class which is used by uNature . Can be accesed by "UNTerrain.terrainData".	153
uNature.Core.Terrains.UNTerrainEditor	156
uNature.Core.Sectors.UNTerrainSector	
An sector class dedicated only for terrains.	156

uNature.Core.Threading.UNThreadManager	
This class handles the multi-threading mechanics.	159
uNature.Core.Terrains.UNTreePrototype	
A custom class for the normal tree prototypes. Holds custom data that is used over this certain terrain data.	161
uNature.Core.Vector2i	164
uNature.Core.Utility.Vector3_XZ_FAST	165
uNature.Core.FoliageClasses.WindSettings	165
uNature.Core.FoliageClasses.WindZones	165

Chapter 4

Namespace Documentation

4.1 uNature Namespace Reference

Namespaces

4.2 uNature.Core Namespace Reference

Namespaces

Classes

- interface [IUTCPhysicsIgnored](#)
Ignore all physics on this script.
- class **UNMath**
An custom math class.
- class **UNPhysics**
This class handles all custom physics.
- struct [UNPhysicsHit_Grass](#)
A class that holds the data for the hit data
- class [UNPhysicsHitsArray](#)
An custom array that holds all ray results in an array
- struct [UNPhysicsObject](#)
This is a base class for a UCPhysicsObject. Every class that inherits this class will be counted in the physics system.
- struct [UNPhysicsTemplate](#)
- struct [Vector2i](#)

4.3 uNature.Core.ClassExtensions Namespace Reference

Classes

- class **ClassExtensions**
Some extensions that helps achieving things that aren't built in with unity.

4.4 uNature.Core.Collections Namespace Reference

Classes

- class [UNDimensionalList](#)
A 2 dimensional list which is used by certain mechanics in [uNature](#).
- class [UNList](#)
A custom list which is used on some important interfaces in UN.

4.5 uNature.Core.Editor Namespace Reference

Namespaces

4.6 uNature.Core.Editor.Helpers Namespace Reference

Classes

- class [UNEditorHelpers](#)

4.7 uNature.Core.Extensions Namespace Reference

Classes

- class [MethodHelperAttribute](#)
- class [UN_ForgeNetworking](#)
- class [UN_GAIA](#)
- class [UN_GENA](#)
- class [UN_MapMagic](#)
- class [UN_PhotonBolt](#)
- class [UN_PhotonCloud](#)
- class [UN_TerrainComposer](#)
- class [UN_UFPS](#)
- class [UN_UNet](#)
- class [UN_WorldStreamer](#)
- class [UNExtension](#)
A uConstruct extension that will allow other 3d party systems to work with uConstruct.
- class [UNExtensionsEditor](#)

4.8 uNature.Core.FoliageClasses Namespace Reference

Classes

- class [BaseInteraction](#)
- class [FoliageChunk](#)
- class [FoliageCore_Chunk](#)
- class [FoliageCore_MainManager](#)
- class [FoliageCore_Sector](#)
- class [FoliageDB](#)

The database class of the Foliage, holds a lot of important data such as Foliage prototypes, Foliage map and more.

- class [FoliageDynamicSurface](#)
- struct [FoliageLODLevel](#)
- class [FoliageManagerInstance](#)
- class [FoliageMesh](#)
- class [FoliageMeshInstance](#)
- class [FoliageMeshInstancesGroup](#)
- class [FoliageMeshManager](#)
- class [FoliagePrototype](#)
- class [FoliageReceiver](#)
- class [FoliageSector](#)

An sector class dedicated only to Foliage.

- class [GPUMesh](#)

A class used to hold the gpu meshes

- class [GPUMeshLOD](#)

GPU Mesh Lods.

- class [PaintBrush](#)
- class **[RenderingPipelineUtility](#)**

The rendering pipe line utility for uNature. Internal Only.

- class [RenderingQueue](#)

An rendering queue.

- class [RenderingQueueInstance](#)

An rendering queue instance.

- struct [RenderingQueueMeshInstanceSimulator](#)

An imposter struct to simulate an instance.

- class [RenderingQueueReceiver](#)

An object that can handle an object queue.

- class [TouchBending](#)
- class [TouchBendingEditor](#)
- class [UNFoliageEditor](#)
- class [UNFoliageManagerEditor](#)
- class [UNMeshData](#)
- class [WindSettings](#)
- class [WindZones](#)

Enumerations

- enum **[FoliageType](#)** { **Prefab**, **Texture** }
- enum **[ShaderType](#)** { **NaN** = 0, **Basic** = 1, **Advanced** = 2, **Custom** = 3 }
- enum **[FoliageResolutions](#)** {
 _128 = 128, **_256** = 256, **_512** = 512, **_1024** = 1024,
 _2048 = 2048 }
- enum **[FoliageGenerationRadius](#)** { **_1x1** = 1, **_3x3** = 3, **_5x5** = 5 }
- enum **[CurrentPaintMethod](#)** { **Normal_Paint**, **Spline_Paint** }

Functions

- delegate void **OnFoliageEnableChanged** ([FoliagePrototype](#) changedPrototype, bool value)
- delegate void **OnFoliageManagerAssigned** ([FoliageCore_MainManager](#) instance)

4.9 uNature.Core.Math Namespace Reference

Classes

- class **UNMath**

4.10 uNature.Core.Networking Namespace Reference

Classes

- class [BaseUNNetworkData](#)
- class [UNNetworkData](#)

A class which can be used for an abstract networking data.

Template Parameters

T	<i>The network connection type which the networking library uses.</i>
---	---

- class [UNNetworkManager](#)

A template for networking, which can be used by networking extensions to easily get the networking actions done.

Template Parameters

T1	<i>the targeted networking connection</i>
T2	<i>the type of the data</i>

- class [UNNetworkPlayerController](#)

Enumerations

- enum **PacketType** { **HealthUpdate** }

4.11 uNature.Core.Pooling Namespace Reference

Classes

- class [HarvestableTIPoolItem](#)

A [Pool](#) item for terrain where the tree instances should be harvestable. (Tree cutting for instance) Inherite from this class to create your own harvestable type.

- interface [IHarvestableItem](#)
- interface [IPoolComponent](#)

- class [Pool](#)
A class that manages the [Pooling](#) of the system, Which allows huge runtime performance increase.
- class [PoolItem](#)
An abstract class that handles the [Pool](#) items.
- class [TerrainPoolItem](#)
A [Pool](#) item for terrain. (Tree instances)

Functions

- delegate void **OnHealthChanged** (int value)
- delegate void **OnItemStateChanged** ([HarvestableTIPoolItem](#) item)
- delegate void **OnItemDamaged** ([HarvestableTIPoolItem](#) item, int damage)
- delegate void **OnTreeInstanceStateChanged** (Terrain terrain, int instanceID)

4.12 uNature.Core.Sectors Namespace Reference

Classes

- class [Chunk](#)
part of the sector which contains information.
- class [ChunkObject](#)
- class [GrassLODLevel](#)
A class that holds a level which all assigned on different frames.
- class [Sector](#)
A sector which is used to divide the UNTerrain objects in the world to increase performance (can handle more than 200k trees!!)
- class [TICChunk](#)
- struct [TreeFetchingTask_MultiThreaded](#)
- class [UNTerrainSector](#)
An sector class dedicated only for terrains.

Functions

- delegate void **SectorRecalculated** (List< [Chunk](#) > newChunks, Vector2 newChunkSize)

4.13 uNature.Core.Seekers Namespace Reference

Classes

- class [UNSeeker](#)
[Seekers](#) are basically GameObjects in the scene which should interact with the objects in the game.
- class [UNSeekerEditor](#)

4.14 uNature.Core.Settings Namespace Reference

Classes

- class [UNSetting](#)
A class which should be used on custom classes that needs to be shown and serialized as a setting.
- class [UNSettingAttribute](#)
The attribute of each setting which handles the drawing of the setting (generically).
- class [UNSettingCategory](#)
The class of a category which handles keeping hold of all of the categories and makes all of the reflection needed.
- class [UNSettings](#)
A class which handles certain settings of aspects in UN
- class [UNSettingsEditor](#)

Enumerations

- enum [UNSettingCategories](#) {
Terrain, General, Networking, Interaction,
Threading, Grass }
The categories of the settings which will be used on the editor.

4.14.1 Enumeration Type Documentation

4.14.1.1 enum `uNature.Core.Settings.UNSettingCategories` [strong]

The categories of the settings which will be used on the editor.

4.15 uNature.Core.Targets Namespace Reference

Classes

- class [UNTarget](#)
A target is what will be taken into account with the system. For example terrains.
- class [UNTargetEditor](#)

4.16 uNature.Core.Terrains Namespace Reference

Classes

- class [UNTerrain](#)
A class that needs to be on each terrain that needs to be taken into account when managing the system.
- class [UNTerrainData](#)
The terrain data class which is used by [uNature](#). Can be accesed by "UNTerrain.terrainData".
- class [UNTerrainEditor](#)
- class [UNTreePrototype](#)
A custom class for the normal tree prototypes. Holds custom data that is used over this certain terrain data.

Enumerations

- enum **TerrainChangedFlags** {
NoChange = 0, **Heightmap** = 1, **TreeInstances** = 2, **DelayedHeightmapUpdate** = 4,
FlushEverythingImmediately = 8, **RemoveDirtyDetailsImmediately** = 16, **TreePrototypesChanged** = 32,
WillBeDestroyed = 256 }
- enum **TerrainTabs** { **Grids**, **Pool**, **Vegetation**, **Trees** }

4.17 uNature.Core.Threading Namespace Reference

Classes

- interface [IThreadTask](#)
A thread task interface. Implement on any customely created thread task.
- class [ThreadItem](#)
This class handles assigning parameters before multi-threaded actions that can be called from outside of unity's main thread. for example : position.
- class [ThreadTask](#)
A thread task that takes no parameters.
- class [UNThreadManager](#)
This class handles the multi-threading mechanics.

Enumerations

- enum **uNature_Thread_Workers** {
One_Worker = 1, **Two_Workers** = 2, **Three_Workers** = 3, **Four_Workers** = 4,
Five_Workers = 5 }

4.18 uNature.Core.Utility Namespace Reference

Classes

- class [BasePrototypeItem](#)
An base prototype item which is used for the [uNature](#) ui utility.
- class [FoliageGrassMap](#)
Channels:
- class [FoliageHeightMap](#)
Channels: R: Heights Channel #1 G: Heights Channel #2
- class [FoliageNormalMap](#)
Channels: R: Heights Channel #1 G: Heights Channel #2
- class [FoliageWorldMaps](#)
- class [UN_TerrainTexturePrototype](#)
The prototype of the terrain texture. [Used for listing the paintable surfaces]
- class **UNBatchUtility**
An utility class for batching items.
- class [UNBrushUtility](#)
Using this class you can paint an brush on the scene.
- struct [UNCombineInstance](#)

- class [UNDictionary](#)
- class [UNFastList](#)
 - A list that requires you to set*
- class [UNMap](#)
 - The abstract Map class.*
- class **UNMapGenerators**
- class **UNStandaloneUtility**
- struct [Vector3_XZ_FAST](#)

4.19 uNature.Demo Namespace Reference

Classes

- class [UN_FirstPersonController](#)
- class [UN_MouseLook](#)

4.20 uNature.Wrappers Namespace Reference

Namespaces

4.21 uNature.Wrappers.Linq Namespace Reference

Classes

- class **Check**
- interface [IOrderedEnumerable](#)
- class **LinqWrapper**
- class [OrderedEnumerable](#)
- class [OrderedSequence](#)
- class [QuickSort](#)
- class [SortContext](#)
- class [SortSequenceContext](#)

Enumerations

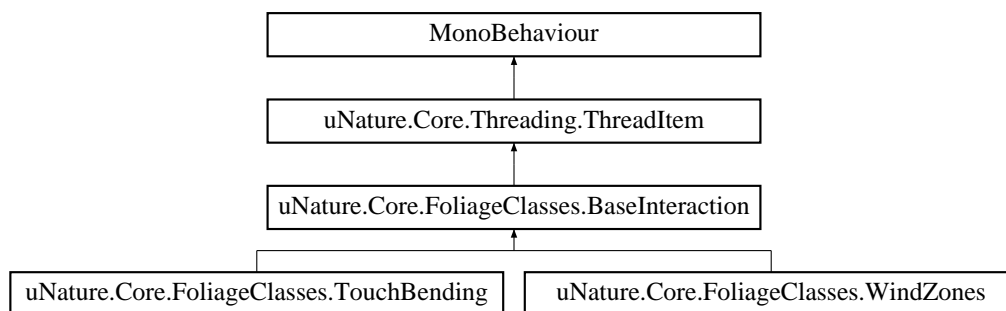
- enum **SortDirection** { **Ascending**, **Descending** }

Chapter 5

Class Documentation

5.1 uNature.Core.FoliageClasses.BaseInteraction Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.BaseInteraction:



Public Member Functions

- void `UpdateInteraction` (`FoliageReceiver` receiver)
Update the interaction

Static Public Member Functions

- static List< `BaseInteraction` > `GetRelevantInteractions` (`FoliageReceiver` receiver)

Protected Member Functions

- override void `OnEnable` ()
- override void `OnDisable` ()
Called when the object is disabled
- override void `OnPositionChanged` (`Vector3` newPosition)
Called when the item's position changed
- virtual void `UpdateInteraction` (`FoliageReceiver` receiver, `Vector2` normalizedPosition)
Please dont use map.SetPixels, it will be assigned later on automatically..
- `Vector2` `GetNormalizedPosition` (`FoliageReceiver` receiver)
- `Vector2` `GetNormalizedPosition` (`Vector2` position, `FoliageReceiver` receiver)

Properties

- static List< [BaseInteraction](#) > **interactions** [get]
- virtual bool **includedInInteractionMap** [get]

Additional Inherited Members

5.1.1 Member Function Documentation

5.1.1.1 **override void uNature.Core.FoliageClasses.BaseInteraction.OnDisable ()** [protected], [virtual]

Called when the object is disabled

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

Reimplemented in [uNature.Core.FoliageClasses.TouchBending](#).

5.1.1.2 **override void uNature.Core.FoliageClasses.BaseInteraction.OnPositionChanged (Vector3 *newPosition*)**
[protected], [virtual]

Called when the item's position changed

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

Reimplemented in [uNature.Core.FoliageClasses.TouchBending](#).

5.1.1.3 **void uNature.Core.FoliageClasses.BaseInteraction.UpdateInteraction (FoliageReceiver *receiver*)**

Update the interaction

Parameters

<i>receiver</i>	
-----------------	--

5.1.1.4 **virtual void uNature.Core.FoliageClasses.BaseInteraction.UpdateInteraction (FoliageReceiver *receiver*, Vector2 *normalizedPosition*)** [protected], [virtual]

Please dont use map.SetPixels, it will be assigned later on automatically..

Parameters

<i>receiver</i>	
-----------------	--

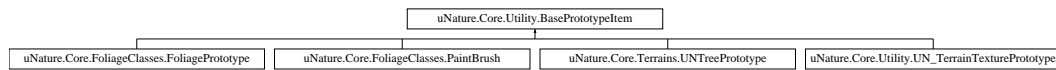
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/Interaction/BaseInteraction.cs

5.2 uNature.Core.Utility.BasePrototypeItem Class Reference

An base prototype item which is used for the [uNature](#) ui utility.

Inheritance diagram for uNature.Core.Utility.BasePrototypeItem:



Protected Member Functions

- virtual Texture2D **GetPreview** ()

Properties

- Texture2D **preview** [get, set]
- virtual bool **isEnabled** [get]
- virtual bool **chooseableOnDisabled** [get]

5.2.1 Detailed Description

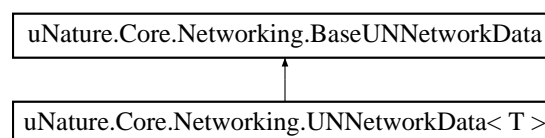
An base prototype item which is used for the [uNature](#) ui utility.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/BasePropertyItem.cs

5.3 uNature.Core.Networking.BaseUNNetworkData Class Reference

Inheritance diagram for uNature.Core.Networking.BaseUNNetworkData:



Public Member Functions

- virtual void **UnPack** ()
Unpack the data

Public Attributes

- int **treeInstanceID**
- string **terrainID**
- PacketType **eventType** = PacketType.HealthUpdate

Protected Attributes

- int **_health**

Properties

- int **minHealth** [get, set]
- int **maxHealth** [get, set]
- int **health** [get, set]

5.3.1 Member Function Documentation

5.3.1.1 virtual void uNature.Core.Networking.BaseUNNetworkData.UnPack () [virtual]

Unpack the data

Reimplemented in [uNature.Core.Networking.UNNetworkData< T >](#).

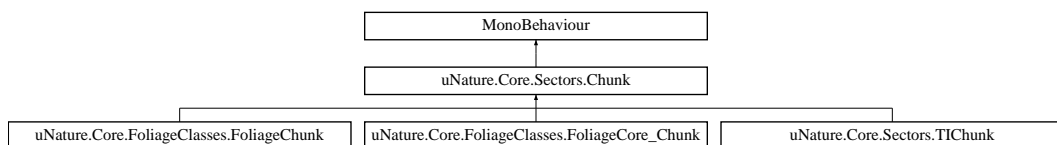
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/BaseUNNetworkData.cs

5.4 uNature.Core.Sectors.Chunk Class Reference

part of the sector which contains information.

Inheritance diagram for uNature.Core.Sectors.Chunk:



Public Member Functions

- virtual void [Awake](#) ()
Called when the object is created/ initializes.

Parameters

terrain	the terrain
---------	-------------

- bool [Contains](#) (Vector2 point, float offset)
Is this point inside the chunk?
- bool [Contains](#) (Vector3 point, float offset)
Is this point inside the chunk?
- virtual void [OnCreated](#) ()
Called when the chunk is created.
- virtual void [ResetChunk](#) ()
Reset the chunk's propoties.
- virtual void [OnDrawGizmos](#) ()
Draw gizmos.

Static Public Member Functions

- static T [CreateChunk](#)< T > (Sector sector, Vector2 position, int x, int z, Vector2 scale, short chunkID)
Create a new chunk

Public Attributes

- Vector2 **terrainRelativeSize**
- Vector2 **minPoint**
- Vector2 **maxPoint**
- short **chunkID**
- int **x**
- int **z**
- Transform **sectorOwner**

Protected Member Functions

- virtual void [OnSizeChanged](#) ()
On size parameter changed.
- virtual void [OnEnable](#) ()
Called on disable
- virtual void [OnDisable](#) ()
Called on disable

Protected Attributes

- Vector3 **_position3D**

Properties

- Vector2 **position** [get, set]
- Vector3 **position3D** [get]
- Vector2 **size** [get, set]
- Vector2 **extents** [get, set]
- Vector2 **center** [get]
- virtual string **chunkType** [get]

5.4.1 Detailed Description

part of the sector which contains information.

5.4.2 Member Function Documentation

5.4.2.1 virtual void uNature.Core.Sectors.Chunk.Awake () [virtual]

Called when the object is created/ initializes.

Parameters

<i>terrain</i>	the terrain
----------------	-------------

Reimplemented in [uNature.Core.FoliageClasses.FoliageCore_Chunk](#), and [uNature.Core.Sectors.TIChunk](#).

5.4.2.2 bool uNature.Core.Sectors.Chunk.Contains (Vector2 *point*, float *offset*)

Is this point inside the chunk?

Parameters

<i>point</i>	the point
--------------	-----------

Returns

5.4.2.3 bool uNature.Core.Sectors.Chunk.Contains (Vector3 *point*, float *offset*)

Is this point inside the chunk?

Parameters

<i>point</i>	the point
--------------	-----------

Returns

5.4.2.4 static T uNature.Core.Sectors.Chunk.CreateChunk< T > (Sector *sector*, Vector2 *position*, int *x*, int *z*, Vector2 *scale*, short *chunkID*) [static]

Create a new chunk

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>sector</i>	
<i>position</i>	
<i>scale</i>	
<i>unTerrain</i>	

Returns

Type Constraints

T* : *Chunk

5.4.2.5 virtual void uNature.Core.Sectors.Chunk.OnCreated () [virtual]

Called when the chunk is created.

Reimplemented in [uNature.Core.Sectors.TIChunk](#), [uNature.Core.FoliageClasses.FoliageCore_Chunk](#), and [uNature.Core.FoliageClasses.FoliageChunk](#).

5.4.2.6 virtual void uNature.Core.Sectors.Chunk.OnDisable () [protected],[virtual]

Called on disable

Reimplemented in [uNature.Core.FoliageClasses.FoliageChunk](#).

5.4.2.7 virtual void uNature.Core.Sectors.Chunk.OnDrawGizmos () [virtual]

Draw gizmos.

Reimplemented in [uNature.Core.Sectors.TIChunk](#), and [uNature.Core.FoliageClasses.FoliageCore_Chunk](#).

5.4.2.8 virtual void uNature.Core.Sectors.Chunk.OnEnable () [protected],[virtual]

Called on disable

5.4.2.9 virtual void uNature.Core.Sectors.Chunk.OnSizeChanged () [protected],[virtual]

On size parameter changed.

Reimplemented in [uNature.Core.FoliageClasses.FoliageCore_Chunk](#), [uNature.Core.Sectors.TIChunk](#), and [uNature.Core.FoliageClasses.FoliageChunk](#).

5.4.2.10 virtual void uNature.Core.Sectors.Chunk.ResetChunk () [virtual]

Reset the chunk's propoties.

Reimplemented in [uNature.Core.Sectors.TIChunk](#).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Chunk.cs

5.5 uNature.Core.Sectors.ChunkObject Class Reference

Public Member Functions

- **ChunkObject** (int _instanceID, TreeInstance treeInstance, Vector3 terrainSize, TerrainData tData, Vector3 terrainPosition)
- void **Remove** ()

Public Attributes

- TreeInstance **treeInstance**
- int **prototypeID**
- int **instanceID**
- Vector3 **worldPosition**
- Vector2 **depthPosition**
- System.DateTime **removedTime**
- float **originalHeight**
- [HarvestableTIPoolItem](#) **prefabHarvestableComponent**
- [HarvestableTIPoolItem](#) **harvestableComponent**

Properties

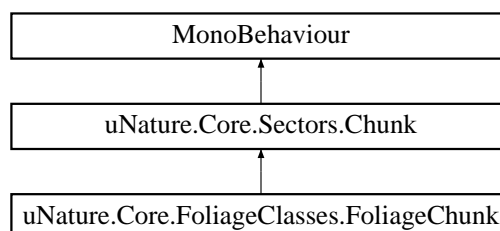
- bool **isRemoved** [get]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Chunk.cs

5.6 uNature.Core.FoliageClasses.FoliageChunk Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageChunk:



Public Member Functions

- override void [OnCreated](#) ()
Create colliders

Protected Member Functions

- override void [OnDisable](#) ()
Called on disable
- override void [OnSizeChanged](#) ()
Called when the size of the chunk is changed.

Additional Inherited Members

5.6.1 Member Function Documentation

5.6.1.1 override void uNature.Core.FoliageClasses.FoliageChunk.OnCreated () [virtual]

Create colliders

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.6.1.2 override void uNature.Core.FoliageClasses.FoliageChunk.OnDisable () [protected],[virtual]

Called on disable

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.6.1.3 override void uNature.Core.FoliageClasses.FoliageChunk.OnSizeChanged () [protected],[virtual]

Called when the size of the chunk is changed.

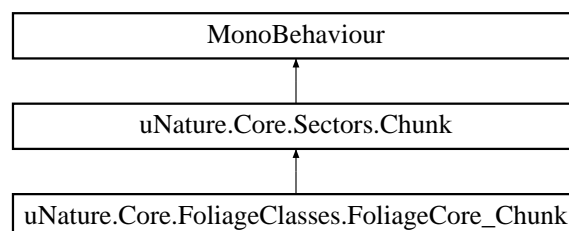
Reimplemented from [uNature.Core.Sectors.Chunk](#).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageChunk.cs

5.7 uNature.Core.FoliageClasses.FoliageCore_Chunk Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageCore_Chunk:



Public Member Functions

- override void [OnDrawGizmos](#) ()
On Draw Gizmos
- override void [OnCreated](#) ()
Called when the chunk is created.
- override void [Awake](#) ()
Called when the object is created/ initializes.

Parameters

terrain	the terrain
---------	-------------

- [FoliageManagerInstance](#) [GetOrCreateFoliageManagerInstance](#) ()
Get the attached Foliage Manager Instance and if not available, create one.
- bool [InBounds](#) (Vector3 normalizedPosition, float distance)
Check for in bounds

Protected Member Functions

- override void [OnSizeChanged](#) ()
On size parameter changed.

Properties

- bool [isFoliageInstanceAttached](#) [get]
Checks if an foliage instance exist on this chunk.

Additional Inherited Members

5.7.1 Member Function Documentation

5.7.1.1 override void [uNature.Core.FoliageClasses.FoliageCore_Chunk.Awake](#) () [virtual]

Called when the object is created/ initializes.

Parameters

<i>terrain</i>	the terrain
----------------	-------------

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.7.1.2 [FoliageManagerInstance](#) [uNature.Core.FoliageClasses.FoliageCore_Chunk.GetOrCreateFoliageManagerInstance](#) ()

Get the attached Foliage Manager Instance and if not available, create one.

Returns

5.7.1.3 `bool uNature.Core.FoliageClasses.FoliageCore_Chunk.InBounds (Vector3 normalizedPosition, float distance)`

Check for in bounds

Parameters

<i>normalizedPosition</i>	
<i>distance</i>	

Returns

5.7.1.4 `override void uNature.Core.FoliageClasses.FoliageCore_Chunk.OnCreated () [virtual]`

Called when the chunk is created.

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.7.1.5 `override void uNature.Core.FoliageClasses.FoliageCore_Chunk.OnDrawGizmos () [virtual]`

On Draw Gizmos

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.7.1.6 `override void uNature.Core.FoliageClasses.FoliageCore_Chunk.OnSizeChanged () [protected], [virtual]`

On size parameter changed.

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.7.2 Property Documentation

5.7.2.1 `bool uNature.Core.FoliageClasses.FoliageCore_Chunk.isFoliageInstanceAttached [get]`

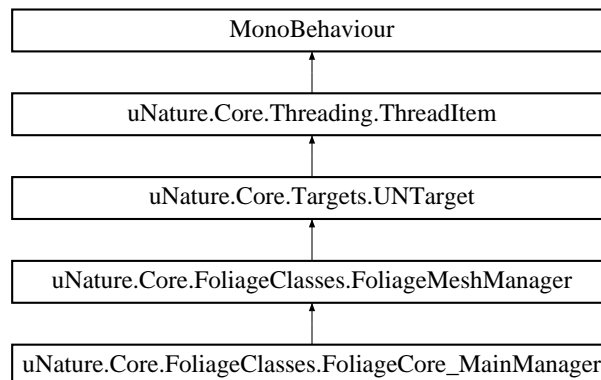
Checks if an foliage instance exist on this chunk.

The documentation for this class was generated from the following file:

- `D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageCore_Chunk.cs`

5.8 uNature.Core.FoliageClasses.FoliageCore_MainManager Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageCore_MainManager:



Public Member Functions

- int [GetChunkID](#) (float x, float z)
Get chunk from bounds.
- bool [CheckChunkInBounds](#) (int chunkID)
Check if the chunk id is in range
- void [InsertFoliageFromTerrain](#) (Terrain terrain, bool saveImmediately)
Copy the terrain's details and use it with the custom Foliage system.
- void [UpdateHeightsOnTerrain](#) (Terrain terrain)
Update the heights on a terrain
- void [UpdateHeights](#) (int x, int z, int scaleX, int scaleZ)
Update Heights On Cords
- byte[,] [GetDetailLayer](#) (int worldX, int worldZ, int sizeX, int sizeZ, int prototypeIndex)
Set detail layer in world cords
- void [SetDetailLayer](#) (int worldX, int worldZ, int sizeX, int sizeZ, int prototypeIndex, byte[,] densities)
Set detail layer in world cords

Static Public Member Functions

- static void [SaveDelayedMaps](#) ()
Save maps that have been marked as delayed (waiting for update)
- static void [RemoveGrassMap](#) (FoliagePrototype prototype)
Remove Grass Map Globally
- static void [WarmUpGrassMaps](#) ()
Update the existing grass maps
- static void [WarmUpGrassMaps](#) (FoliageCore_Chunk[] specificChunks)
Update the existing grass maps
- static void [ResetGrassMap](#) (List< FoliagePrototype > prototypes)
Reset the existing grass maps
- static void [InitializeAndCreatelfNotFound](#) ()
Create an instance if not created
- static void [DestroyManager](#) ()
Destroy this manager instance and clean up the data.

Public Attributes

- const int **FOLIAGE_MAIN_AREA_RADIUS** = 10240
- const int **FOLIAGE_INSTANCE_AREA_SIZE** = (FOLIAGE_MAIN_AREA_RADIUS * 2) / FOLIAGE_MAIN←
_AREA_RESOLUTION
- bool **useQualitySettingsShadowDistance** = false
- float **foliageShadowDistance** = 100

Protected Member Functions

- override void **Awake** ()
Initiate awake settings.
- override void **OnEnable** ()
- override void **OnDisable** ()
Called when the object is disabled
- override void **Update** ()
Update...

Properties

- static **FoliageCore_MainManager** **instance** [get]
- string **guid** [get]
- new bool **enabled** [get, set]
- float **density** [get, set]
- int **FoliageGenerationLayerMask** [get, set]
- **FoliageCore_Sector** **sector** [get]

Events

- static OnFoliageManagerAssigned **OnFoliageManagerAssignedEvent**

Additional Inherited Members

5.8.1 Member Function Documentation

5.8.1.1 override void uNature.Core.FoliageClasses.FoliageCore_MainManager.Awake () [protected],[virtual]

Initiate awake settings.

Reimplemented from [uNature.Core.Targets.UNTarget](#).

5.8.1.2 bool uNature.Core.FoliageClasses.FoliageCore_MainManager.CheckChunkInBounds (int *chunkID*)

Check if the chunk id is in range

Parameters

<i>chunkID</i>	
----------------	--

Returns

5.8.1.3 `static void uNature.Core.FoliageClasses.FoliageCore_MainManager.DestroyManager () [static]`

Destroy this manager instance and clean up the data.

5.8.1.4 `int uNature.Core.FoliageClasses.FoliageCore_MainManager.GetChunkID (float x, float z)`

Get chunk from bounds.

[REMOVE MAIN MANAGER POSITION FROM CORDS!!] for example: `cordX = transform.position.x - FoliageCore_MainManager.instance.transform.position.x`.

Parameters

<i>x</i>	
<i>z</i>	

Returns

5.8.1.5 `byte [,] uNature.Core.FoliageClasses.FoliageCore_MainManager.GetDetailLayer (int worldX, int worldZ, int sizeX, int sizeZ, int prototypeIndex)`

Set detail layer in world cords

Parameters

<i>worldX</i>	WORLD CORDS!!
<i>worldZ</i>	WORLD CORDS!!
<i>sizeX</i>	WORLD CORDS!!
<i>sizeZ</i>	WORLD CORDS!!
<i>prototypeIndex</i>	prototype.id

5.8.1.6 `static void uNature.Core.FoliageClasses.FoliageCore_MainManager.InitializeAndCreatelfNotFound () [static]`

Create an instance if not created

5.8.1.7 void uNature.Core.FoliageClasses.FoliageCore_MainManager.InsertFoliageFromTerrain (Terrain *terrain*, bool *saveImmediately*)

Copy the terrain's details and use it with the custom Foliage system.

Parameters

<i>terrain</i>	
----------------	--

5.8.1.8 override void uNature.Core.FoliageClasses.FoliageCore_MainManager.OnDisable () [protected],
[virtual]

Called when the object is disabled

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.8.1.9 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.RemoveGrassMap (FoliagePrototype *prototype*) [static]

Remove Grass Map Globally

Parameters

<i>id</i>	
-----------	--

5.8.1.10 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.ResetGrassMap (List< FoliagePrototype > *prototypes*) [static]

Reset the existing grass maps

Parameters

<i>prototype</i>	
------------------	--

5.8.1.11 static void uNature.Core.FoliageClasses.FoliageCore_MainManager.SaveDelayedMaps () [static]

Save maps that have been marked as delayed (waiting for update)

5.8.1.12 void uNature.Core.FoliageClasses.FoliageCore_MainManager.SetDetailLayer (int *worldX*, int *worldZ*, int *sizeX*, int *sizeZ*, int *prototypeIndex*, byte *densities*[,])

Set detail layer in world cords

Parameters

<i>worldX</i>	WORLD CORDS!!
<i>worldZ</i>	WORLD CORDS!!
<i>sizeX</i>	WORLD CORDS!!
<i>sizeZ</i>	WORLD CORDS!!
<i>prototypeIndex</i>	prototype.id
<i>densities</i>	the density in bytes from 0 -> 15

5.8.1.13 `override void uNature.Core.FoliageClasses.FoliageCore_MainManager.Update ()` [protected],
[virtual]

Update...

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.8.1.14 `void uNature.Core.FoliageClasses.FoliageCore_MainManager.UpdateHeights (int x, int z, int scaleX, int scaleZ)`

Update Heights On Cords

Parameters

<i>x</i>	
<i>z</i>	
<i>scaleX</i>	
<i>scaleZ</i>	

5.8.1.15 `void uNature.Core.FoliageClasses.FoliageCore_MainManager.UpdateHeightsOnTerrain (Terrain terrain)`

Update the heights on a terrain

Parameters

<i>terrain</i>	
----------------	--

5.8.1.16 `static void uNature.Core.FoliageClasses.FoliageCore_MainManager.WarmUpGrassMaps ()` [static]

Update the existing grass maps

Parameters

<i>prototype</i>	
------------------	--

5.8.1.17 `static void uNature.Core.FoliageClasses.FoliageCore_MainManager.WarmUpGrassMaps (FoliageCore_Chunk[] specificChunks) [static]`

Update the existing grass maps

Parameters

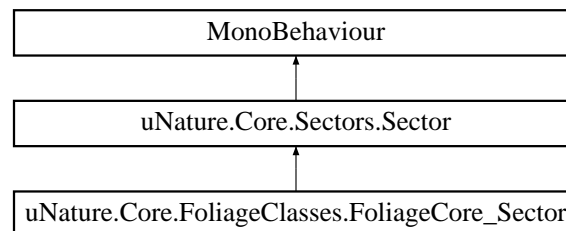
<i>prototype</i>	
------------------	--

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageCore_MainManager.cs

5.9 uNature.Core.FoliageClasses.FoliageCore_Sector Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageCore_Sector:



Public Attributes

- List< [FoliageCore_Chunk](#) > **foliageChunks** = new List<[FoliageCore_Chunk](#)>()

Protected Member Functions

- override void [OnChunkCreated](#) ([Chunk](#) chunk)
Called when a chunk is created to allow custom logic on the inherited sectors.
- override void [OnStartCreatingChunks](#) ()
Called right before starting to create the chunks.
- override void [OnResolutionChanged](#) ()
Called when the resolution has been updated.

Additional Inherited Members

5.9.1 Member Function Documentation

5.9.1.1 `override void uNature.Core.FoliageClasses.FoliageCore_Sector.OnChunkCreated (Chunk chunk) [protected], [virtual]`

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

<code>chunk</code>	
--------------------	--

Reimplemented from [uNature.Core.Sectors.Sector](#).

5.9.1.2 `override void uNature.Core.FoliageClasses.FoliageCore_Sector.OnResolutionChanged () [protected], [virtual]`

Called when the resolution has been updated.

Reimplemented from [uNature.Core.Sectors.Sector](#).

5.9.1.3 `override void uNature.Core.FoliageClasses.FoliageCore_Sector.OnStartCreatingChunks () [protected], [virtual]`

Called right before starting to create the chunks.

Reimplemented from [uNature.Core.Sectors.Sector](#).

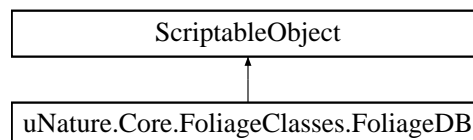
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foliage/GPU_Uilities/FoliageCore_Sector.cs

5.10 uNature.Core.FoliageClasses.FoliageDB Class Reference

The database class of the Foliage, holds a lot of important data such as Foliage prototypes, Foliage map and more.

Inheritance diagram for uNature.Core.FoliageClasses.FoliageDB:



Public Member Functions

- [FoliagePrototype AddPrototype](#) (Texture2D texture, GameObject prefab, float minWidth, float minHeight, float maxWidth, float maxHeight, float spread, int layer, Color healthyColor, Color dryColor)
Add a new Foliage prototype.
- [FoliagePrototype AddPrototype](#) (DetailPrototype detailPrototype)
Add a new Foliage prototype.
- [FoliagePrototype AddPrototype](#) (Texture2D texture)
Add a new Foliage prototype.
- [FoliagePrototype AddPrototype](#) (GameObject prefab)
Add a new Foliage prototype.
- void [RemovePrototype](#) ([FoliagePrototype](#) prototype)
Remove an existing Foliage prototype.
- void [UpdateShaderWindSettings](#) ()
Update wind settings globally
- void [UpdateShaderGeneralSettings](#) ()
This will update the general settings of the shader such as density, min width, max width etc

Public Attributes

- `WindSettings` **globalWindSettings** = new `WindSettings()`

Properties

- static `FoliageDB` **instance** [get]
Get the instance, if not found, it will automatically create one.
- static List< `FoliagePrototype` > **unSortedPrototypes** [get]
- static Dictionary< int, `FoliagePrototype` > **sortedPrototypes** [get]
- List< `PaintBrush` > **brushes** [get, set]

5.10.1 Detailed Description

The database class of the Foliage, holds a lot of important data such as Foliage prototypes, Foliage map and more.

5.10.2 Member Function Documentation

5.10.2.1 FoliagePrototype `uNature.Core.FoliageClasses.FoliageDB.AddPrototype (Texture2D texture, GameObject prefab, float minWidth, float minHeight, float maxWidth, float maxHeight, float spread, int layer, Color healthyColor, Color dryColor)`

Add a new Foliage prototype.

5.10.2.2 FoliagePrototype `uNature.Core.FoliageClasses.FoliageDB.AddPrototype (DetailPrototype detailPrototype)`

Add a new Foliage prototype.

5.10.2.3 FoliagePrototype `uNature.Core.FoliageClasses.FoliageDB.AddPrototype (Texture2D texture)`

Add a new Foliage prototype.

5.10.2.4 FoliagePrototype `uNature.Core.FoliageClasses.FoliageDB.AddPrototype (GameObject prefab)`

Add a new Foliage prototype.

5.10.2.5 void `uNature.Core.FoliageClasses.FoliageDB.RemovePrototype (FoliagePrototype prototype)`

Remove an existing Foliage prototype.

5.10.2.6 void `uNature.Core.FoliageClasses.FoliageDB.UpdateShaderGeneralSettings ()`

This will update the general settings of the shader such as density, min width, max width etc

5.10.2.7 void uNature.Core.FoliageClasses.FoliageDB.UpdateShaderWindSettings ()

Update wind settings globally

5.10.3 Property Documentation

5.10.3.1 FoliageDB uNature.Core.FoliageClasses.FoliageDB.instance [static],[get]

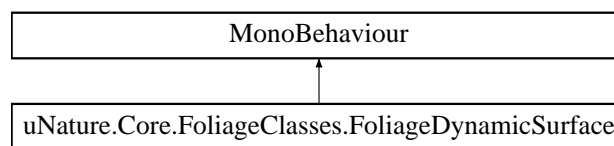
Get the instance, if not found, it will automatically create one.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageDB.cs

5.11 uNature.Core.FoliageClasses.FoliageDynamicSurface Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageDynamicSurface:



Public Attributes

- float **updateDistanceDifference** = 0.05f

Protected Member Functions

- virtual void **OnEnable** ()
- virtual void **OnDisable** ()
- virtual void **Update** ()
- virtual void **ApplyPositionChange** ()
- virtual void **ApplyScaleChange** ()

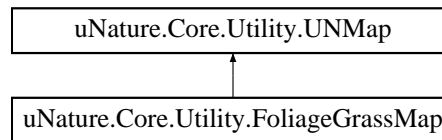
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Utilities/FoliageDynamicSurface.cs

5.12 uNature.Core.Utility.FoliageGrassMap Class Reference

Channels:

Inheritance diagram for uNature.Core.Utility.FoliageGrassMap:



Public Member Functions

- **FoliageGrassMap** (Texture2D texture, [FoliagePrototype](#) prototype, [FoliageManagerInstance](#) mInstance)
- **FoliageGrassMap** (Texture2D texture, Color32[] pixels, [FoliagePrototype](#) prototype, [FoliageManagerInstance](#) mInstance)
- void **UpdateMap** ()
- void **ResetDensity** ()
- Texture2D [SafeGetMap](#) ([FoliagePrototype](#) prototype)
This method will make sure that the map exists and if it doesnt it will create it.
- void [Dispose](#) ()
Destroy this current grass map.
- byte [GetDensity](#) (int x, int z)
Get density at normalized x & z
- void [SetDensity](#) (int x, int z, byte density)
Set density at normalized x & z
- void [SetDensityFast](#) (int index, byte density)
- void [MarkDensitiesDirty](#) ()
Mark the densities as dirty.
- void **Save** ()

Static Public Member Functions

- static void **SaveAllMaps** ()
- static void **ApplyAreaSizeChange** ([FoliageManagerInstance](#) mInstance)
- static void [UpdateGrassMaps](#) ([FoliageManagerInstance](#) mInstance)
Update all of the available grass maps (pixels)

Properties

- float **perlinScale** [get, set]
- int **prototypeID** [get]
- static bool **globalDirty** [get]

Additional Inherited Members

5.12.1 Detailed Description

Channels:

R: Free G: Free B: Density A: Perlin Noise

5.12.2 Member Function Documentation

5.12.2.1 void uNature.Core.Utility.FoliageGrassMap.Dispose ()

Destroy this current grass map.

5.12.2.2 byte uNature.Core.Utility.FoliageGrassMap.GetDensity (int x, int z)

Get density at normalized x & z

Parameters

x	
z	

Returns

5.12.2.3 void uNature.Core.Utility.FoliageGrassMap.MarkDensitiesDirty ()

Mark the densities as dirty.

5.12.2.4 Texture2D uNature.Core.Utility.FoliageGrassMap.SafeGetMap (FoliagePrototype *prototype*)

This method will make sure that the map exists and if it doesnt it will create it.

Used mainly for the rendering so if the user accidently/ purposely removed the grass map it will automatically generate a new one so it wont affect the system.

Parameters

<i>manager</i>	
----------------	--

Returns

5.12.2.5 void uNature.Core.Utility.FoliageGrassMap.SetDensity (int *x*, int *z*, byte *density*)

Set density at normalized x & z

Parameters

<i>x</i>	
<i>z</i>	

Returns

5.12.2.6 void uNature.Core.Utility.FoliageGrassMap.SetDensityFast (int *index*, byte *density*)

Parameters

<i>x</i>	
<i>z</i>	

Returns

5.12.2.7 static void uNature.Core.Utility.FoliageGrassMap.UpdateGrassMaps (*FoliageManagerInstance mInstance*)
[static]

Update all of the availble grass maps (pixels)

Parameters

<i>size</i>	
-------------	--

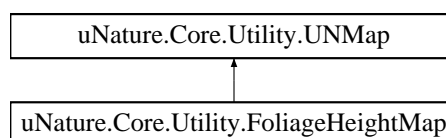
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.13 uNature.Core.Utility.FoliageHeightMap Class Reference

Channels: R: Heights Channel #1 G: Heights Channel #2

Inheritance diagram for uNature.Core.Utility.FoliageHeightMap:



Public Member Functions

- **FoliageHeightMap** (Texture2D texture, [FoliageManagerInstance](#) mInstance)
- void **Save** ()

Additional Inherited Members

5.13.1 Detailed Description

Channels: R: Heights Channel #1 G: Heights Channel #2

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.14 uNature.Core.FoliageClasses.FoliageLODLevel Struct Reference

Public Member Functions

- **FoliageLODLevel** (float lodDistance, float lodValue)

Public Attributes

- const int **LOD_MAX_DISTANCE** = 500
- Vector2 **_vectorRepresentation**
- bool **_isDirty**

Properties

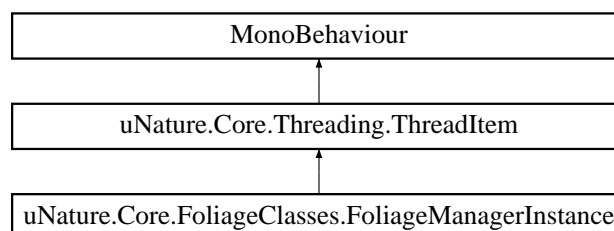
- Vector2 **vectorRepresentation** [get]
- float **lodDistance** [get, set]
- float **lodValue** [get, set]

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

5.15 uNature.Core.FoliageClasses.FoliageManagerInstance Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageManagerInstance:



Public Member Functions

- void [ForceMapsRestore](#) ()
Restores the changes on the maps as long as saving changes on runtime isnt checked on the settings.
- void **UpdateMaterialBlock** (MaterialPropertyBlock mBlock)
- void **RemoveGrassMap** ([FoliagePrototype](#) prototype)
- int [TransformCord](#) (float x, float removeOffset)
Transform 1 cord
- float [TransformCordFloat](#) (float x, float removeOffset)
Transform 1 cord
- int [TransformCordCustom](#) (float x, float removeOffset, float multiplier)
Transform 1 cord
- float [TransformCordCustomFloat](#) (float x, float removeOffset, float multiplier)
Transform 1 cord
- int [InverseCord](#) (float x, float addOffset)
Transform 1 cord
- float [InverseCordFloat](#) (float x, float addOffset)
Transform 1 cord
- int [InverseCordCustom](#) (float x, float addOffset, float multiplier)
Transform 1 cord
- float [InverseCordCustomFloat](#) (float x, float addOffset, float multiplier)
Transform 1 cord

Static Public Member Functions

- static void **CleanUp** ([FoliageManagerInstance](#) mInstance)

Protected Member Functions

- override void **OnEnable** ()
- override void [OnDisable](#) ()
Called when the object is disabled
- virtual void **UpdateResolutionChange** ()

Properties

- static List< [FoliageManagerInstance](#) > **instances** [get]
- string **guid** [get]
- Dictionary< int, [GPU Mesh](#) > **meshInstances** [get]
- Vector3 **pos** [get]
- FoliageResolutions **foliageAreaResolution** [get, set]
- int **foliageAreaResolutionIntegral** [get]
- float **transformCordsMultiplier** [get, set]
- int **FoliageGenerationLayerMask** [get, set]
- Texture2D **colorMap** [get, set]
- [FoliageWorldMaps](#) **worldMaps** [get, set]
- Dictionary< int, [FoliageGrassMap](#) > **grassMaps** [get]
- Dictionary< int, [FoliageGrassMap](#) > **unSafeGrassMaps** [get]
- [FoliageCore_Chunk](#) **attachedTo** [get]

Additional Inherited Members

5.15.1 Member Function Documentation

5.15.1.1 `void uNature.Core.FoliageClasses.FoliageManagerInstance.ForceMapsRestore ()`

Restores the changes on the maps as long as saving changes on runtime isnt checked on the settings.

5.15.1.2 `int uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCord (float x, float addOffset)`

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

5.15.1.3 `int uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCordCustom (float x, float addOffset, float multiplier)`

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

5.15.1.4 `float uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCordCustomFloat (float x, float addOffset, float multiplier)`

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

5.15.1.5 float uNature.Core.FoliageClasses.FoliageManagerInstance.InverseCordFloat (float *x*, float *addOffset*)

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

5.15.1.6 override void uNature.Core.FoliageClasses.FoliageManagerInstance.OnDisable () [protected], [virtual]

Called when the object is disabled

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.15.1.7 int uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCord (float *x*, float *removeOffset*)

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

5.15.1.8 int uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCordCustom (float *x*, float *removeOffset*, float *multiplier*)

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

5.15.1.9 float uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCordCustomFloat (float *x*, float *removeOffset*, float *multiplier*)

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

5.15.1.10 float uNature.Core.FoliageClasses.FoliageManagerInstance.TransformCordFloat (float *x*, float *removeOffset*)

Transform 1 cord

Parameters

<i>x</i>	
----------	--

Returns

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageManagerInstance.cs

5.16 uNature.Core.FoliageClasses.FoliageMesh Class Reference

Public Member Functions

- **FoliageMesh** (GameObject go, int layer, string name, Vector3 offset)

Public Attributes

- const int **OPTIMIZATION_MESH_INSTANCES_DENSITIES_LIMITER** = 12
- Mesh **mesh**
- Material **mat**
- Vector3 **eulerAngles**
- Vector3 **scale** = Vector3.one
- Vector3 **offset**
- int **vertexCount**

Properties

- Vector3 **rendererScale** [get]
- Vector3 **worldScale** [get]
- UNMeshData **meshData** [get]
- ShaderType **shaderType** [get]
- int **MeshInstancesLimiter_Optimization_Clamp** [get]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

5.17 uNature.Core.FoliageClasses.FoliageMeshInstance Class Reference

Public Member Functions

- Vector3 **GetPosition** (Vector3 pos)
- byte **GetDensity** (float positionX, float positionZ, byte maxDensity, Color32[] mapPixels, int mapWidth)
get density of the mesh instance.

Static Public Member Functions

- static Mesh[][] **CreateFoliageInstances** (int prototypeIndex, List< byte > densities, out [FoliageMeshInstancesGroup](#) meshGroup, FoliageResolutions resolution)
Generates the mesh instances.
- static void **MarkDensitiesDirty** ()
Mark all of the densities as dirty

Public Attributes

- [FoliagePrototype](#) **prototype**
- Vector3 **position**
- Mesh **mesh**
- Bounds **cullBounds**
- int **boundsSizeX**
- int **boundsSizeZ**
- Vector3 **boundsExtents**
- int **meshIndex**

Properties

- [FoliageChunk](#) **currentChunk** [get, set]

5.17.1 Member Function Documentation

5.17.1.1 `static Mesh [][] uNature.Core.FoliageClasses.FoliageMeshInstance.CreateFoliageInstances (int prototypeIndex, List< byte > densities, out FoliageMeshInstancesGroup meshGroup, FoliageResolutions resolution) [static]`

Generates the mesh instances.

Returns

5.17.1.2 `byte uNature.Core.FoliageClasses.FoliageMeshInstance.GetDensity (float positionX, float positionZ, byte maxDensity, Color32[] mapPixels, int mapWidth)`

get density of the mesh instance.

Parameters

<i>pos</i>	
------------	--

Returns

5.17.1.3 `static void uNature.Core.FoliageClasses.FoliageMeshInstance.MarkDensitiesDirty () [static]`

Mark all of the densities as dirty

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageMeshInstance.cs

5.18 uNature.Core.FoliageClasses.FoliageMeshInstancesGroup Class Reference

Public Member Functions

- void **Add** ([FoliageMeshInstance](#) instance)
- void **Finish** ()
- void **Destroy** ()

Public Attributes

- [FoliageMeshInstance](#)[] **meshInstances** = null

Properties

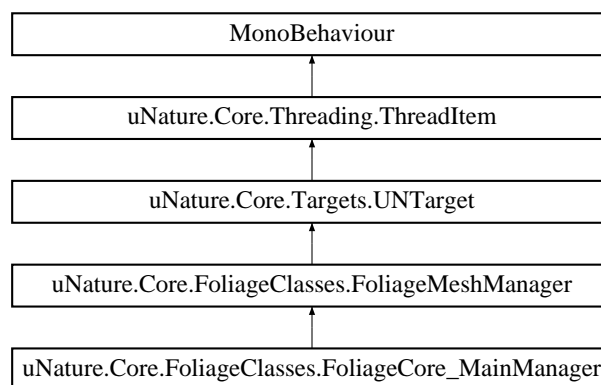
- int **Count** [get]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageMeshManager.cs

5.19 uNature.Core.FoliageClasses.FoliageMeshManager Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageMeshManager:



Public Member Functions

- void [UpdateMeshBounds](#) (Vector3 centerPos)
Update the mesh instances bounds
- void **OnGlobalPostRender** (Camera camera)
- void **DEBUG_DrawUI** ()

Static Public Member Functions

- static void [GenerateFoliageMeshInstances](#) ()
Generate new mesh instances
- static void [GenerateFoliageMeshInstances](#) (FoliageResolutions resolution)
Generate new mesh instances
- static void [GenerateFoliageMeshInstances](#) (int prototypeID)
Generate new mesh instances
- static void [GenerateFoliageMeshInstanceForIndex](#) (int prototypeIndex, FoliageResolutions resolution)
Create Foliage mesh instances for a certain index and foliage size.
- static void [DestroyMeshInstance](#) (int prototypeID)
Destroy a mesh instance
- static void [RegenerateQueueInstances](#) ()
Restart all of the queue instances.
- static Mesh [CreateNewMesh](#) ()
Create a new mesh instace

Public Attributes

- bool **DEBUG_Window_Open** = false
- bool **DEBUG_Window_Minimized** = false

Static Public Attributes

- static List< Mesh > **globalMeshesThreshold** = new List<Mesh>()

Protected Member Functions

- override void **OnEnable** ()
- override void **OnDisable** ()
Called when the object is disabled
- override void **Update** ()
Update...

Static Protected Member Functions

- static void **DestroyMeshInstances** ()
Destroy the current mesh instances.

Protected Attributes

- int **_lastRenderedVertices**
- int **_lastRenderedDrawCalls**
- int **_lastRenderedPrototypes**

Properties

- static int **PROPERTY_ID_WORLDPOSITION** [get]
- static int **PROPERTY_ID_GRASSMAP** [get]
- static int **PROPERTY_ID_FOLIAGE_INTERACTION_TOUCH_BENDING_OBJECTS** [get]
- static Dictionary< FoliageResolutions, Dictionary< int, **GPUMesh** > > **prototypeMeshInstances** [get]
- MaterialPropertyBlock **propertyBlock** [get]
- int **lastRenderedVertices** [get]
- int **lastRenderedDrawCalls** [get]
- int **lastRenderedPrototypes** [get]

5.19.1 Member Function Documentation

5.19.1.1 static Mesh uNature.Core.FoliageClasses.FoliageMeshManager.CreateNewMesh () [static]

Create a new mesh instace

Returns

5.19.1.2 static void uNature.Core.FoliageClasses.FoliageMeshManager.DestroyMeshInstance (int *prototypeID*) [static]

Destroy a mesh instance

Parameters

<i>prototypeID</i>	
--------------------	--

5.19.1.3 `static void uNature.Core.FoliageClasses.FoliageMeshManager.DestroyMeshInstances () [static], [protected]`

Destroy the current mesh instances.

5.19.1.4 `static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstanceForIndex (int prototypeIndex, FoliageResolutions resolution) [static]`

Create Foliage mesh instances for a certain index and foliage size.

Parameters

<i>meshInstances</i>	
<i>prototypeIndex</i>	

5.19.1.5 `static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstances () [static]`

Generate new mesh instances

Parameters

<i>areaSize</i>	
-----------------	--

5.19.1.6 `static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstances (FoliageResolutions resolution) [static]`

Generate new mesh instances

Parameters

<i>areaSize</i>	
-----------------	--

5.19.1.7 `static void uNature.Core.FoliageClasses.FoliageMeshManager.GenerateFoliageMeshInstances (int prototypeID) [static]`

Generate new mesh instances

Parameters

<i>areaSize</i>	
-----------------	--

5.19.1.8 `override void uNature.Core.FoliageClasses.FoliageMeshManager.OnDisable ()` [protected],[virtual]

Called when the object is disabled

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.19.1.9 `static void uNature.Core.FoliageClasses.FoliageMeshManager.RegenerateQueueInstances ()` [static]

Restart all of the queue instances.

5.19.1.10 `override void uNature.Core.FoliageClasses.FoliageMeshManager.Update ()` [protected],[virtual]

Update...

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.19.1.11 `void uNature.Core.FoliageClasses.FoliageMeshManager.UpdateMeshBounds (Vector3 centerPos)`

Update the mesh instances bounds

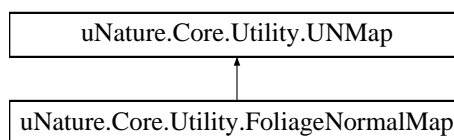
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foliage/GPU_Uilities/FoliageMeshManager.cs

5.20 uNature.Core.Utility.FoliageNormalMap Class Reference

Channels: R: Heights Channel #1 G: Heights Channel #2

Inheritance diagram for uNature.Core.Utility.FoliageNormalMap:



Public Member Functions

- **FoliageNormalMap** (Texture2D texture, [FoliageManagerInstance](#) mInstance)
- void [Save](#) ()
Save

Additional Inherited Members

5.20.1 Detailed Description

Channels: R: Heights Channel #1 G: Heights Channel #2

5.20.2 Member Function Documentation

5.20.2.1 void uNature.Core.Utility.FoliageNormalMap.Save ()

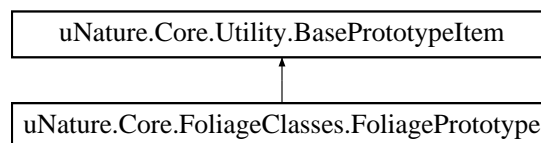
Save

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.21 uNature.Core.FoliageClasses.FoliagePrototype Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliagePrototype:



Public Member Functions

- bool **EqualsToPrototype** (DetailPrototype detail)
- void **ApplyWind** ()
Apply the wind parameters to this Foliage prototype.
- void **ApplyColorMap** (Texture2D map, Texture2D normalMap)
Apply color map
- void **ApplyGrassMap** (Texture2D map)
Apply color map
- void **UpdateManagerInformation** ()
Update the global spread noise.

Static Public Member Functions

- static **FoliagePrototype CreatePrototype** (Texture2D texture, GameObject prefab, float minWidth, float min↵ Height, float maxWidth, float maxHeight, float spread, int layer, int id, Color healthyColor, Color dryColor)
Create a prototype.

Public Attributes

- const string **SHADER_BASIC_NAME** = "uNature/FoliageShader_Basic"
- const string **SHADER_ADVANCED_NAME** = "uNature/FoliageShader_Advanced"
- const float **SIZE_MIN_VALUE** = 0.1f
- const float **SIZE_MAX_VALUE** = 5.0f
- **WindSettings** customWindSettings = new **WindSettings**()
- Vector3 **instancedEuler**

Static Public Attributes

- static Color **DEFAULT_HEALTHY_COLOR** = new Color(33f / 255, 129f / 255, 25f / 255, 1)
- static Color **DEFAULT_DRY_COLOR** = new Color(205f / 255, 188f / 255, 26f / 255, 1)

Protected Member Functions

- override Texture2D [GetPreview](#) ()
Get Preview

Properties

- static GameObject **FoliageTexGameObject** [get]
- FoliageType **FoliageType** [get, set]
- GameObject **FoliageMesh** [get, set]
- Texture2D **FoliageTexture** [get, set]
- int **id** [get]
- int **maxFoliageCapability** [get]
- float **spread** [get, set]
- float **minimumWidth** [get, set]
- float **maximumWidth** [get, set]
- float **minimumHeight** [get, set]
- float **maximumHeight** [get, set]
- bool **receiveShadows** [get, set]
- Color **dryColor** [get, set]
- Color **healthyColor** [get, set]
- bool **castShadows** [get, set]
- int **fadeDistance** [get, set]
- int **maxGeneratedDensity** [get, set]
- bool **useColorMap** [get, set]
- bool **rotateNormals** [get, set]
- string **name** [get]
- bool **enabled** [get, set]
- int **meshLodsCount** [get]
- int **renderingLayer** [get, set]
- bool **touchBendingEnabled** [get, set]
- float **touchBendingStrength** [get, set]
- float **cutOff** [get, set]
- override bool **isEnabled** [get]
- override bool **chooseableOnDisabled** [get]
- bool **useCustomWind** [get, set]
- bool **useLODs** [get, set]
- [FoliageLODLevel](#)[] **lods** [get, set]
- [FoliageMesh](#) **FoliageInstancedMeshData** [get]
- List< byte > **densitiesLODs** [get]

Events

- static OnFoliageEnableChanged **OnFoliageEnabledStateChangedEvent**

5.21.1 Member Function Documentation

5.21.1.1 void uNature.Core.FoliageClasses.FoliagePrototype.ApplyColorMap (Texture2D *map*, Texture2D *normalMap*)

Apply color map

Res = area size.

5.21.1.2 void uNature.Core.FoliageClasses.FoliagePrototype.ApplyGrassMap (Texture2D *map*)

Apply color map

Res = area size.

5.21.1.3 void uNature.Core.FoliageClasses.FoliagePrototype.ApplyWind ()

Apply the wind parameters to this Foliage prototype.

5.21.1.4 static FoliagePrototype uNature.Core.FoliageClasses.FoliagePrototype.CreatePrototype (Texture2D *texture*, GameObject *prefab*, float *minWidth*, float *minHeight*, float *maxWidth*, float *maxHeight*, float *spread*, int *layer*, int *id*, Color *healthyColor*, Color *dryColor*) [static]

Create a prototype.

Parameters

<i>texture</i>	
<i>prefab</i>	
<i>minSize</i>	
<i>maxSize</i>	
<i>spread</i>	
<i>id</i>	

Returns

5.21.1.5 override Texture2D uNature.Core.FoliageClasses.FoliagePrototype.GetPreview () [protected], [virtual]

Get Preview

Returns

Reimplemented from [uNature.Core.Utility.BasePrototypeItem](#).

5.21.1.6 void uNature.Core.FoliageClasses.FoliagePrototype.UpdateManagerInformation ()

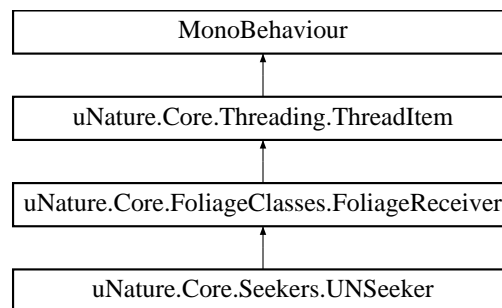
Update the global spread noise.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

5.22 uNature.Core.FoliageClasses.FoliageReceiver Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.FoliageReceiver:



Public Attributes

- bool **isGrassReceiver** = true

Static Public Attributes

- static readonly List< [FoliageReceiver](#) > **FReceivers** = new List<[FoliageReceiver](#)>()

Protected Member Functions

- override void **OnEnable** ()
- override void [OnDisable](#) ()
Called when the object is disabled
- override void [Update](#) ()
Update...

Protected Attributes

- float **_grassCheckDistance** = 20f

Properties

- [FoliageCore_Chunk](#)[] **neighbors** [get]
- [FoliageCore_Chunk](#) **middleFoliageChunkFromNeighbors** [get]
- float **grassCheckDistance** [get, set]
- Camera **playerCamera** [get, set]
- [RenderingQueueReceiver](#) **queueInstance** [get, set]

Additional Inherited Members

5.22.1 Member Function Documentation

5.22.1.1 `override void uNature.Core.FoliageClasses.FoliageReceiver.OnDisable ()` [protected],[virtual]

Called when the object is disabled

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.22.1.2 `override void uNature.Core.FoliageClasses.FoliageReceiver.Update ()` [protected],[virtual]

Update...

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

Reimplemented in [uNature.Core.Seekers.UNSeeker](#).

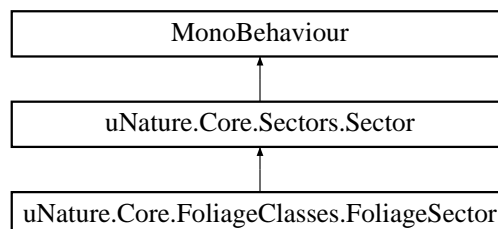
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageReceiver.cs

5.23 uNature.Core.FoliageClasses.FoliageSector Class Reference

An sector class dedicated only to Foliage.

Inheritance diagram for uNature.Core.FoliageClasses.FoliageSector:



Public Attributes

- List< [FoliageChunk](#) > **FoliageChunks** = new List<[FoliageChunk](#)>()

Protected Member Functions

- override void [OnChunkCreated](#) ([Chunk](#) chunk)
Called when a chunk is created to allow custom logic on the inherited sectors.
- override void [OnStartCreatingChunks](#) ()
Called right before starting to create the chunks.

Additional Inherited Members

5.23.1 Detailed Description

An sector class dedicated only to Foliage.

5.23.2 Member Function Documentation

5.23.2.1 override void [uNature.Core.FoliageClasses.FoliageSector.OnChunkCreated](#) ([Chunk](#) *chunk*) [protected],
[virtual]

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

chunk	
-----------------------	--

Reimplemented from [uNature.Core.Sectors.Sector](#).

5.23.2.2 override void [uNature.Core.FoliageClasses.FoliageSector.OnStartCreatingChunks](#) () [protected],
[virtual]

Called right before starting to create the chunks.

Reimplemented from [uNature.Core.Sectors.Sector](#).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageSector.cs

5.24 uNature.Core.Utility.FoliageWorldMaps Class Reference

Public Member Functions

- **FoliageWorldMaps** (Texture2D heightMap, Texture2D normalMap, [FoliageManagerInstance](#) mInstance)
- Vector2 [NormalizeHeight](#) (float worldHeight)
Normalize a world height into a converted height.
- void [UpdateHeightsAndNormals](#) (float x, float z, int sizeX, int sizeZ, bool save)

- Update height on a certain range.*
- void [UpdateHeightsAndNormals](#) (bool save)
- Update the heights and normals all over the map.*
- void [UpdateHeightAndNormal](#) (int index, float height)
- Update height*
- void [UpdateHeightAndNormal](#) (int index, Vector3 normal)
- Update normal*
- void [UpdateHeightAndNormal](#) (int index, float height, Vector3 normal)
- Update height and normal*
- float [GetHeight](#) (int index)
- Get an height on the height map.*
- void [Save](#) ()
- Save the world maps*
- void [SetPixels32Delayed](#) ()
- Set pixels on the world maps, delayed.*
- void [RestoreChanges](#) ()
- Reset the changes of runtime.*
- override bool **Equals** (object obj)
- override int **GetHashCode** ()

Static Public Member Functions

- static Vector3 [TransformNormals](#) (Vector3 normal)
- Normalize the world normals into converted normals.*
- static float [ClampNegativeIntoPositive](#) (float value)
- Convert point from negative to positive.*
- static void [ReGenerateGlobally](#) ()
- Regenerate all of the world maps.*
- static void [SaveAllMaps](#) ()
- Save all of the world maps.*
- static void **ApplyAreaSizeChange** ([FoliageManagerInstance](#) mInstance)

Properties

- [FoliageHeightMap](#) **heightMap** [get, set]
- [FoliageNormalMap](#) **normalMap** [get, set]
- bool **dirty** [get]
- static bool **globalDirty** [get]

5.24.1 Member Function Documentation

5.24.1.1 static float uNature.Core.Utility.FoliageWorldMaps.ClampNegativeIntoPositive (float *value*) [static]

Convert point from negative to positive.

-1 = 0; 0 = 0.5; 1 = 1;

Parameters

<i>value</i>	
--------------	--

Returns**5.24.1.2 float uNature.Core.Utility.FoliageWorldMaps.GetHeight (int *index*)**

Get an height on the height map.

Parameters

<i>pixel</i>	
--------------	--

Returns**5.24.1.3 Vector2 uNature.Core.Utility.FoliageWorldMaps.NormalizeHeight (float *worldHeight*)**

Normalize a world height into a converted height.

Parameters

<i>worldHeight</i>	
--------------------	--

Returns**5.24.1.4 static void uNature.Core.Utility.FoliageWorldMaps.ReGenerateGlobally () [static]**

Regenerate all of the world maps.

5.24.1.5 void uNature.Core.Utility.FoliageWorldMaps.RestoreChanges ()

Reset the changes of runtime.

5.24.1.6 void uNature.Core.Utility.FoliageWorldMaps.Save ()

Save the world maps

5.24.1.7 `static void uNature.Core.Utility.FoliageWorldMaps.SaveAllMaps () [static]`

Save all of the world maps.

5.24.1.8 `void uNature.Core.Utility.FoliageWorldMaps.SetPixels32Delayed ()`

Set pixels on the world maps, delayed.

5.24.1.9 `static Vector3 uNature.Core.Utility.FoliageWorldMaps.TransformNormals (Vector3 normal) [static]`

Normalize the world normals into converted normals.

Parameters

<i>normal</i>	
---------------	--

Returns

5.24.1.10 `void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightAndNormal (int index, float height)`

Update height

Parameters

<i>worldMap</i>	
-----------------	--

5.24.1.11 `void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightAndNormal (int index, Vector3 normal)`

Update normal

Parameters

<i>worldMap</i>	
-----------------	--

5.24.1.12 `void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightAndNormal (int index, float height, Vector3 normal)`

Update height and normal

Parameters

<i>worldMap</i>	
-----------------	--

5.24.1.13 void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightsAndNormals (float x, float z, int sizeX, int sizeZ, bool save)

Update height on a certain range.

Parameters

x	
z	
sizeX	
sizeZ	

5.24.1.14 void uNature.Core.Utility.FoliageWorldMaps.UpdateHeightsAndNormals (bool save)

Update the heights and normals all over the map.

Parameters

save	
------	--

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.25 uNature.Core.FoliageClasses.GPUMesh Class Reference

A class used to hold the gpu meshes

Public Member Functions

- **GPUMesh** (Mesh[][] LODMeshes, [FoliageMeshInstancesGroup](#) LODMeshInstance, List< byte > LOD↔ Levels, int prototypeIndex, FoliageResolutions resolution)
- void **Destroy** ()
- int **GetMesh** (int density)

Public Attributes

- List< [GPUMeshLOD](#) > **meshes** = new List<[GPUMeshLOD](#)>()
 - [FoliageMeshInstancesGroup](#) **LODMeshInstances** = null
- Dimension 1 : x chunk Dimension 2 : z chunk Dimension 3 : LOD index*

5.25.1 Detailed Description

A class used to hold the gpu meshes

5.25.2 Member Data Documentation

5.25.2.1 FoliageMeshInstancesGroup uNature.Core.FoliageClasses.GPUMesh.LODMeshInstances = null

Dimension 1 : x chunk Dimension 2 : z chunk Dimension 3 : LOD index

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageMeshManager.cs

5.26 uNature.Core.FoliageClasses.GPUMeshLOD Class Reference

GPU Mesh Lods.

Public Member Functions

- **GPUMeshLOD** (Mesh[] _meshes, int _density, int _prototypeIndex)
- void **Destroy** ()

Public Attributes

- Mesh[] **meshes**
- int **density**

5.26.1 Detailed Description

GPU Mesh Lods.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/GPU_Uilities/FoliageMeshManager.cs

5.27 uNature.Core.Sectors.GrassLODLevel Class Reference

A class that holds a level which all assigned on different frames.

Public Member Functions

- void **Add** (int x, int value, Vector2 pos)

Static Public Member Functions

- static [GrassLODLevel](#) **Create** ()

Public Attributes

- `UNDimensionalList< int > details = new UNDimensionalList<int>()`
- `Vector2 position = new Vector2(Mathf.Infinity, Mathf.Infinity)`

5.27.1 Detailed Description

A class that holds a level which all assigned on different frames.

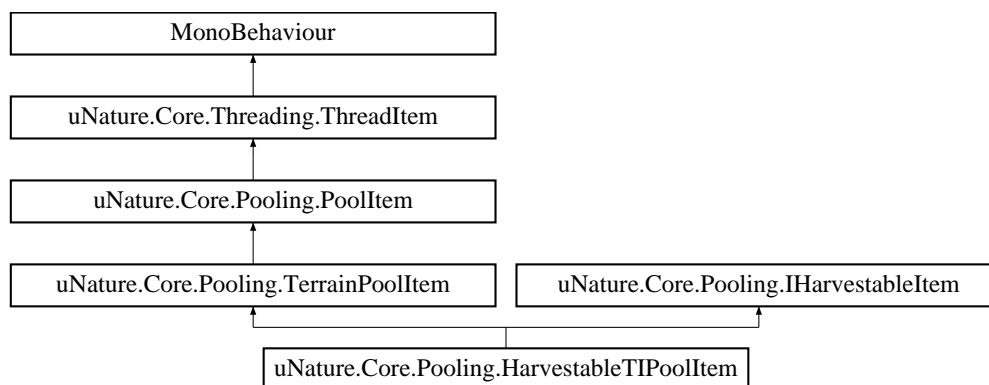
The documentation for this class was generated from the following file:

- `D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Chunk.cs`

5.28 uNature.Core.Pooling.HarvestableTIPoolItem Class Reference

A [Pool](#) item for terrain where the tree instances should be harvestable. (Tree cutting for instance) Inherite from this class to create your own harvestable type.

Inheritance diagram for `uNature.Core.Pooling.HarvestableTIPoolItem`:



Public Member Functions

- override void [Awake](#) ()
Called on awake.
- virtual void [HandleHealthChange](#) (int damage)
Handle the health change, remove the current tree instance if needed, and instantiate a replacment.

Parameters

damage	<i>The amount of damage that the tree has received.</i>
--------	---

- virtual void [HandleColliderDeath](#) ()
Handle death of the colliders (remove tree instance from terrain and replace it with actual tree instance prefab)
- virtual void [HandleTreeInstanceDeath](#) ()
Handle death of an actual tree instance (Add gravity and make it fall)

- virtual void [Hit](#) ()
Hit this harvestable building and apply damage
- virtual void [Hit](#) (int damage)
Hit this harvestable building and apply damage

Parameters

damage	apply the damage
--------	------------------

- override void [OnReturnedToPool](#) ()
Called when the item returns to the [Pool](#), reset the propoties
- override void [OnPool](#) ()
Called when the item pulled to the [Pool](#)

Public Attributes

- int [minHealth](#) = 0
the minimum health possible to be assigned to the tree instance (For example - 0).
- int [maxHealth](#) = 100
The maximum amount of health that can be assigned to this tree instance, which will also be assigned on default (For example - 100).
- float [respawnTimeInMinutes](#) = 2
- float [minFallDisappearTime](#) = 2
- float [maxFallDisappearTime](#) = 10

Static Public Attributes

- static bool [canHarvestCollider](#) = true
Can this machine harvest a COLLIDER ?

Properties

- int [health](#) [get, set]

Events

- static OnItemStateChanged [OnItemPooledEvent](#)
Called when an HarvestableTreeInstance has been Pooled
- static OnItemDamaged [OnItemDamagedEvent](#)
Called when any harvestable item has been damaged.
- static OnItemStateChanged [OnItemReturnedToPoolEvent](#)
Called when an HarvestableTreeInstance has been returned to [Pool](#)
- OnHealthChanged [OnHealthChangedEvent](#)

Additional Inherited Members

5.28.1 Detailed Description

A [Pool](#) item for terrain where the tree instances should be harvestable. (Tree cutting for instance) Inherit from this class to create your own harvestable type.

5.28.2 Member Function Documentation

5.28.2.1 `override void uNature.Core.Pooling.HarvestableTIPoolItem.Awake () [virtual]`

Called on awake.

Reimplemented from [uNature.Core.Pooling.PoolItem](#).

5.28.2.2 `virtual void uNature.Core.Pooling.HarvestableTIPoolItem.HandleColliderDeath () [virtual]`

Handle death of the colliders (remove tree instance from terrain and replace it with actual tree instance prefab)

5.28.2.3 `virtual void uNature.Core.Pooling.HarvestableTIPoolItem.HandleHealthChange (int damage) [virtual]`

Handle the health change, remove the current tree instance if needed, and instantiate a replacment.

Parameters

<i>damage</i>	The amount of damage that the tree has received.
---------------	--

5.28.2.4 `virtual void uNature.Core.Pooling.HarvestableTIPoolItem.HandleTreeInstanceDeath () [virtual]`

Handle death of an actual tree instance (Add gravity and make it fall)

5.28.2.5 `virtual void uNature.Core.Pooling.HarvestableTIPoolItem.Hit () [virtual]`

Hit this harvestable building and apply damage

Implements [uNature.Core.Pooling.IHarvestableItem](#).

5.28.2.6 `virtual void uNature.Core.Pooling.HarvestableTIPoolItem.Hit (int damage) [virtual]`

Hit this harvestable building and apply damage

Parameters

<i>damage</i>	apply the damage
---------------	------------------

Implements [uNature.Core.Pooling.IHarvestableItem](#).

5.28.2.7 `override void uNature.Core.Pooling.HarvestableTIPoolItem.OnPool () [virtual]`

Called when the item pulled to the [Pool](#)

Reimplemented from [uNature.Core.Pooling.PoolItem](#).

5.28.2.8 `override void uNature.Core.Pooling.HarvestableTIPoolItem.OnReturnedToPool () [virtual]`

Called when the item returns to the [Pool](#), reset the propoties

Reimplemented from [uNature.Core.Pooling.PoolItem](#).

5.28.3 Member Data Documentation

5.28.3.1 `bool uNature.Core.Pooling.HarvestableTIPoolItem.canHarvestCollider = true [static]`

Can this machine harvest a COLLIDER ?

5.28.3.2 `int uNature.Core.Pooling.HarvestableTIPoolItem.maxHealth = 100`

The maximum amount of health that can be assigned to this tree instance, which will also be assigned on default (For example - 100).

5.28.3.3 `int uNature.Core.Pooling.HarvestableTIPoolItem.minHealth = 0`

the minimum health possible to be assigned to the tree instance (For example - 0).

5.28.4 Event Documentation

5.28.4.1 `OnItemDamaged uNature.Core.Pooling.HarvestableTIPoolItem.OnItemDamagedEvent [static]`

Called when any harvestable item has been damaged.

5.28.4.2 `OnItemStateChanged uNature.Core.Pooling.HarvestableTIPoolItem.OnItemPooledEvent [static]`

Called when an HarvestableTreeInstance has been Pooled

5.28.4.3 `OnItemStateChanged uNature.Core.Pooling.HarvestableTIPoolItem.OnItemReturnedToPoolEvent [static]`

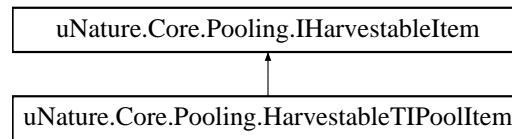
Called when an HarvestableTreeInstance has been returned to [Pool](#)

The documentation for this class was generated from the following file:

- `D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/PoolItems/HarvestableTIPoolItem.cs`

5.29 uNature.Core.Pooling.IHarvestableItem Interface Reference

Inheritance diagram for uNature.Core.Pooling.IHarvestableItem:



Public Member Functions

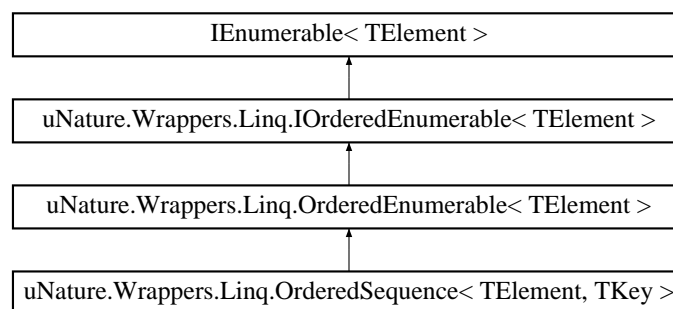
- void **Hit** ()
- void **Hit** (int damage)

The documentation for this interface was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/PoolItems/HarvestableTIPoolItem.cs

5.30 uNature.Wrappers.Linq.IOrderedEnumerable< TElement > Interface Template Reference

Inheritance diagram for uNature.Wrappers.Linq.IOrderedEnumerable< TElement >:



Public Member Functions

- [IOrderedEnumerable< TElement >](#) **CreateOrderedEnumerable**< **TKey** > (Func< TElement, TKey > keySelector, IComparer< TKey > comparer, bool descending)

The documentation for this interface was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.31 uNature.Core.Pooling.IPoolComponent Interface Reference

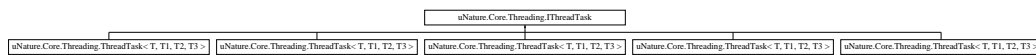
The documentation for this interface was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/Abstracts/IPoolComponent.cs

5.32 uNature.Core.Threading.IThreadTask Interface Reference

A thread task interface. Implement on any customely created thread task.

Inheritance diagram for uNature.Core.Threading.IThreadTask:



Public Member Functions

- void **Invoke** ()

Properties

- int **creationFrame** [get]

5.32.1 Detailed Description

A thread task interface. Implement on any customely created thread task.

The documentation for this interface was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.33 uNature.Core.IUTCPysicsIgnored Interface Reference

Ignore all physics on this script.

Properties

- bool **ignore** [get]

5.33.1 Detailed Description

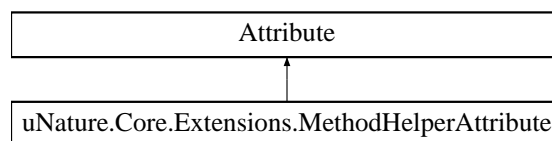
Ignore all physics on this script.

The documentation for this interface was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysics.cs

5.34 uNature.Core.Extensions.MethodHelperAttribute Class Reference

Inheritance diagram for uNature.Core.Extensions.MethodHelperAttribute:

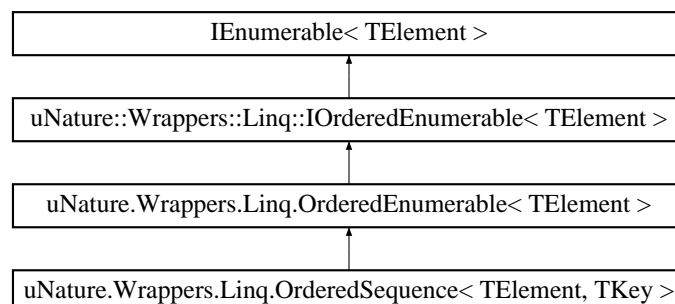


The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Base/UNExtension.cs

5.35 uNature.Wrappers.Linq.OrderedEnumerable< TElement > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.OrderedEnumerable< TElement >:



Public Member Functions

- virtual `IEnumerator< TElement >` **GetEnumerator** ()
- abstract `SortContext< TElement >` **CreateContext** (`SortContext< TElement >` current)
- `IOrderedEnumerable< TElement >` **CreateOrderedEnumerable**< `TKey` > (`Func< TElement, TKey >` selector, `IComparer< TKey >` comparer, `bool` descending)

Protected Member Functions

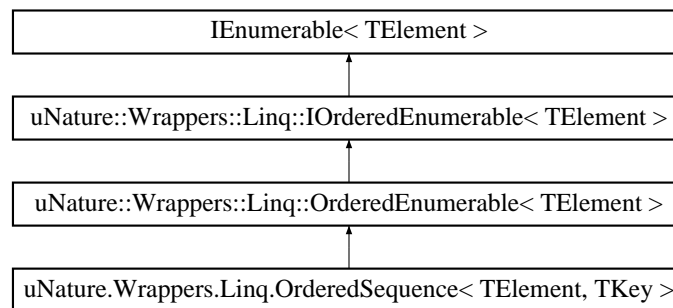
- **OrderedEnumerable** (IEnumerable< TElement > source)
- abstract IEnumerable< TElement > **Sort** (IEnumerable< TElement > source)

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.36 uNature.Wrappers.Linq.OrderedSequence< TElement, TKey > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.OrderedSequence< TElement, TKey >:

**Public Member Functions**

- override IEnumerator< TElement > **GetEnumerator** ()
- override [SortContext](#)< TElement > **CreateContext** ([SortContext](#)< TElement > current)

Protected Member Functions

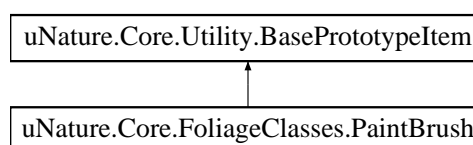
- override IEnumerable< TElement > **Sort** (IEnumerable< TElement > source)

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.37 uNature.Core.FoliageClasses.PaintBrush Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.PaintBrush:



Public Member Functions

- **PaintBrush** (Texture2D _texture)
- void **TryToResize** (int size)

Public Attributes

- Texture2D **brushTexture**

Protected Member Functions

- override Texture2D **GetPreview** ()

Properties

- Texture2D **instancedTexture** [get]
- Color32[,] **pixels** [get]

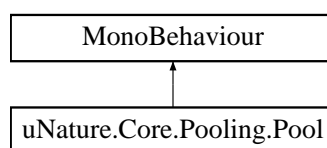
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/PaintBrush.cs

5.38 uNature.Core.Pooling.Pool Class Reference

A class that manages the [Pooling](#) of the system, Which allows huge runtime performance increase.

Inheritance diagram for uNature.Core.Pooling.Pool:



Public Member Functions

- void [AddToPool](#) ([PoolItem](#) item, int itemID, int itemID_Offset)
Add an item to the [Pool](#).
- void [RemoveFromPool](#) ([PoolItem](#) item)
Remove an item from the [Pool](#)
- void [ReturnToPool](#) ([PoolItem](#) item, bool force)
Return a certain item to [Pool](#).
- void [TryResetOnUID](#) (int uid, bool forceReset)
Reset a certain item which is on a certain UID

Parameters

uid	<i>the targeted UID</i>
forceReset	<i>Force reset will make it ignore the locked state of the item.</i>

- T [TryPool< T >](#) (int itemUID, int itemID_Offset, int uid, bool locked)
Try to [Pool](#) an item, will return null if no target is found.
- List< [PoolItem](#) > [GetPoolOfItem](#) (int itemUID, int itemID_Offset)
Get [Pool](#) of a certain item
- void [PoolItem](#) ([PoolItem](#) [PoolItem](#), bool locked, int uid)
[Pool](#) the certain item.
- void [ResetFarAway](#) ()
This method will find and reset far away items to be "recycled"
- bool [IsAlreadyPooled](#) (int uid)
Check if a certain uid is already Pooled.
- T [TryGetType< T >](#) ()
Try to get an object from the [Pool](#) with a certain component.

Static Public Member Functions

- static [Pool](#) [CreatePool](#) (string name, GameObject requester)
Create a new [Pool](#)
- static void [RemoveDuplications](#) (string name)
Remove [Pool](#) duplications.

Public Attributes

- List< [PoolItem](#) > [items](#) = new List<[PoolItem](#)>()
A list that holds all of the [Pool](#) items in our [Pool](#).
- GameObject [owner](#)
Who created this [Pool](#)?

5.38.1 Detailed Description

A class that manages the [Pooling](#) of the system, Which allows huge runtime performance increase.

5.38.2 Member Function Documentation**5.38.2.1 void uNature.Core.Pooling.Pool.AddToPool ([PoolItem](#) item, int itemID, int itemID_Offset)**

Add an item to the [Pool](#).

Parameters

<i>item</i>	the item.
<i>itemID</i>	The targeted item id
<i>itemID_Offset</i>	The offset of the item id to make it unique.

5.38.2.2 `static Pool uNature.Core.Pooling.Pool.CreatePool (string name, GameObject requester) [static]`

Create a new [Pool](#)

Parameters

<i>name</i>	the Pool name (Without Pool at the end)
<i>requester</i>	who is the owner of this Pool

Returns

the newly created [Pool](#).

5.38.2.3 `List<PoolItem> uNature.Core.Pooling.Pool.GetPoolOfItem (int itemUID, int itemID_Offset)`

Get [Pool](#) of a certain item

Parameters

<i>itemUID</i>	
<i>itemID_Offset</i>	

Returns

5.38.2.4 `bool uNature.Core.Pooling.Pool.IsAlreadyPooled (int uid)`

Check if a certain uid is already Pooled.

Parameters

<i>uid</i>	the uid of the targeted item
------------	------------------------------

Returns

is this item already Pooled?

5.38.2.5 `void uNature.Core.Pooling.Pool.PoolItem (PoolItem PoolItem, bool locked, int uid)`

[Pool](#) the certain item.

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>PoolItem</i>	
-----------------	--

5.38.2.6 static void uNature.Core.Pooling.Pool.RemoveDuplications (string *name*) [static]

Remove [Pool](#) duplications.

Parameters

<i>name</i>	
-------------	--

5.38.2.7 void uNature.Core.Pooling.Pool.RemoveFromPool ([PoolItem](#) *item*)

Remove an item from the [Pool](#)

Parameters

<i>item</i>	the item.
-------------	-----------

5.38.2.8 void uNature.Core.Pooling.Pool.ResetFarAway ()

This method will find and reset far away items to be "recycled"

5.38.2.9 void uNature.Core.Pooling.Pool.ReturnToPool ([PoolItem](#) *item*, bool *force*)

Return a certain item to [Pool](#).

Parameters

<i>item</i>	the item.
-------------	-----------

\

Parameters

<i>force</i>	making force true, will make the system ignore the locked state of the item. (if exists)
--------------	--

5.38.2.10 T uNature.Core.Pooling.Pool.TryGetType< T > ()

Try to get an object from the [Pool](#) with a certain component.

Template Parameters

<i>T</i>	the type of the component
----------	---------------------------

Returns

Type Constraints

T : Component

5.38.2.11 `T uNature.Core.Pooling.Pool.TryPool< T > (int itemUID, int itemID_Offset, int uid, bool locked)`

Try to [Pool](#) an item, will return null if no target is found.

Parameters

<i>itemUID</i>	the uid of the item (without offset)
<i>itemID_Offset</i>	the offset of the required item id
<i>uid</i>	a unique id of the object which will be attached to this game object. (HAS TO BE UNIQUE...)
<i>locked</i>	if the Pool item is locked, it wont be able to return to Pool unless its unlocked.

Returns

A [Pool](#) item.

Type Constraints

T : PoolItem

5.38.2.12 `void uNature.Core.Pooling.Pool.TryResetOnUID (int uid, bool forceReset)`

Reset a certain item which is on a certain UID

Parameters

<i>uid</i>	the targeted UID
<i>forceReset</i>	Force reset will make it ignore the locked state of the item.

5.38.3 Member Data Documentation

5.38.3.1 `List<PoolItem> uNature.Core.Pooling.Pool.items = new List<PoolItem>()`

A list that holds all of the [Pool](#) items in our [Pool](#).

5.38.3.2 GameObject uNature.Core.Pooling.Pool.owner

Who created this [Pool](#)?

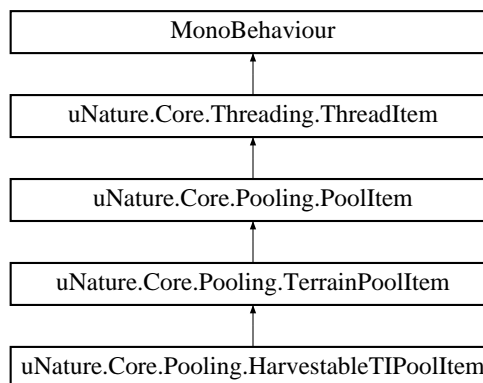
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/Pool.cs

5.39 uNature.Core.Pooling.PoolItem Class Reference

An abstract class that handles the [Pool](#) items.

Inheritance diagram for uNature.Core.Pooling.PoolItem:



Public Member Functions

- virtual void [Awake](#) ()
Called on awake.
- virtual void [OnPool](#) ()
Called when the item has been Pooled.
- virtual void [OnReturnedToPool](#) ()
Called when the item has returned to the [Pool](#)
- virtual void [OnCreated](#) ()
Called when the item has been created.
- virtual void [MoveItem](#) (Vector3 position)
Move the item to a certain position. NOTE: in order to move the item, use this method and DONT change the position externally!!

Public Attributes

- [Pool](#) [Pool](#)
What [Pool](#) are we belonged to?
- `GameObject` [_gameObject](#)
An gameobject reference which can be used on a different thread.
- `bool` [used](#)
is the item currently used?
- `bool` [locked](#)
Is this [Pool](#) item locked? If so, dont let it return back to [Pool](#) unless forced.
- `int` [realItemID](#)
The [Pool](#) item unique id, which is used to identify the item. (not including offset)
- `int` [itemID_Offset](#)
The offset of the item id which allows the item id to be more unique. Can be left 0.
- `int` [uid](#) = -1
What is the uid of the item we are attached to.

Protected Member Functions

- override void [OnEnable](#) ()
Called when the object is enabled.
- override void [OnDisable](#) ()
Called when the object is disabled.

Properties

- static `System.Type[]` [PoolTypes](#) [get]
All the [Pool](#) types in the assembly.
- `int` [itemID](#) [get]
The [Pool](#) item unique id, which is used to identify the item. (including offset)

Additional Inherited Members

5.39.1 Detailed Description

An abstract class that handles the [Pool](#) items.

5.39.2 Member Function Documentation

5.39.2.1 virtual void uNature.Core.Pooling.PoolItem.Awake () [virtual]

Called on awake.

Reimplemented in [uNature.Core.Pooling.HarvestableTIPoolItem](#).

5.39.2.2 virtual void uNature.Core.Pooling.PoolItem.MoveItem (Vector3 position) [virtual]

Move the item to a certain position. NOTE: in order to move the item, use this method and DONT change the position externally!!

Parameters

<i>position</i>	target position.
-----------------	------------------

Reimplemented in [uNature.Core.Pooling.TerrainPoolItem](#).

5.39.2.3 virtual void uNature.Core.Pooling.PoolItem.OnCreated () [virtual]

Called when the item has been created.

5.39.2.4 override void uNature.Core.Pooling.PoolItem.OnDisable () [protected],[virtual]

Called when the object is disabled.

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.39.2.5 override void uNature.Core.Pooling.PoolItem.OnEnable () [protected],[virtual]

Called when the object is enabled.

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.39.2.6 virtual void uNature.Core.Pooling.PoolItem.OnPool () [virtual]

Called when the item has been Pooled.

Reimplemented in [uNature.Core.Pooling.HarvestableTIPoolItem](#).

5.39.2.7 virtual void uNature.Core.Pooling.PoolItem.OnReturnedToPool () [virtual]

Called when the item has returned to the [Pool](#)

Reimplemented in [uNature.Core.Pooling.HarvestableTIPoolItem](#).

5.39.3 Member Data Documentation

5.39.3.1 GameObject uNature.Core.Pooling.PoolItem._gameObject

An gameobject reference which can be used on a different thread.

5.39.3.2 int uNature.Core.Pooling.PoolItem.itemID_Offset

The offset of the item id which allows the item id to be more unique. Can be left 0.

5.39.3.3 `bool uNature.Core.Pooling.PoolItem.locked`

Is this [Pool](#) item locked? If so, dont let it return back to [Pool](#) unless forced.

5.39.3.4 `Pool uNature.Core.Pooling.PoolItem.Pool`

What [Pool](#) are we belonged to?

5.39.3.5 `int uNature.Core.Pooling.PoolItem.realItemID`

The [Pool](#) item unique id, which is used to identify the item. (not including offset)

5.39.3.6 `int uNature.Core.Pooling.PoolItem.uid = -1`

What is the uid of the item we are attached to.

5.39.3.7 `bool uNature.Core.Pooling.PoolItem.used`

is the item currently used?

5.39.4 Property Documentation

5.39.4.1 `int uNature.Core.Pooling.PoolItem.itemID` `[get]`

The [Pool](#) item unique id, which is used to identify the item. (including offset)

5.39.4.2 `System.Type [] uNature.Core.Pooling.PoolItem.PoolTypes` `[static], [get]`

All the [Pool](#) types in the assembly.

The documentation for this class was generated from the following file:

- `D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/Abstracts/PoolItem.cs`

5.40 `uNature.Wrappers.Linq.QuickSort< TElement >` Class Template Reference

Static Public Member Functions

- static `IEnumerable< TElement > Sort` (`IEnumerable< TElement > source`, `SortContext< TElement > context`)

The documentation for this class was generated from the following file:

- `D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs`

5.41 uNature.Core.FoliageClasses.RenderingQueue Class Reference

An rendering queue.

Public Member Functions

- **RenderingQueue** ([FoliageManagerInstance](#) mInstance, [FoliageCore_Chunk](#) mChunk, Vector3 snapPosition, Dictionary< int, [RenderingQueueInstance](#) > fetchedQueueInstance)
- void **DrawDebug** ()
- [UNFastList](#)< [RenderingQueueMeshInstanceSimulator](#) > **GetFromPool** ([FoliageManagerInstance](#) mInstance, [FoliagePrototype](#) prototype)
- void **AddToPool** ([UNFastList](#)< [RenderingQueueMeshInstanceSimulator](#) > obj, [FoliagePrototype](#) prototype)

Public Attributes

- const int **GENERATION_RADIUS** = 3
- const int **GENERATION_RADIUS_OFFSET** = 1
- [FoliageManagerInstance](#) mInstance
- Dictionary< int, [RenderingQueueInstance](#) > queueInstance
- bool **queueInstanceNull** = true

Static Public Attributes

- static Dictionary< int, Stack< [UNFastList](#)< [RenderingQueueMeshInstanceSimulator](#) > > > **RenderingQueueInstancesPool**

5.41.1 Detailed Description

An rendering queue.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.42 uNature.Core.FoliageClasses.RenderingQueueInstance Class Reference

An rendering queue instance.

Public Member Functions

- **RenderingQueueInstance** ([FoliageManagerInstance](#) mInstance, [FoliagePrototype](#) prototype, [GPUMesh](#) gpuMesh, [FoliageGrassMap](#) grassMap)
- void **DrawDebug** ()

Static Public Attributes

- static int **count** = 0

5.42.1 Detailed Description

An rendering queue instance.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.43 uNature.Core.FoliageClasses.RenderingQueueMeshInstanceSimulator Struct Reference

An imposter struct to simulate an instance.

Public Member Functions

- **RenderingQueueMeshInstanceSimulator** (Vector3 position, [FoliageMeshInstance](#) _meshInstance, [FoliageManagerInstance](#) mInstance, [FoliagePrototype](#) _prototype)
- **RenderingQueueMeshInstanceSimulator UpdateDensity** ([GPUMesh](#) gpuMesh, byte maxDensity, [Foliage](#)↔[GrassMap](#) grassMap, int mapWidth, float densityMultiplier)
- void **Render** (MaterialPropertyBlock mBlock, Camera camera, Plane[] cameraPlanes, Vector3 normalized↔CameraPosition, bool isPlaying, bool useQualitySettingsShadow, float shadowsDistance)
- void **DrawDebug** ()

Public Attributes

- float **x**
- float **z**
- float **worldX**
- float **worldZ**

Properties

- byte **density** [get]
- Mesh **mesh** [get]

5.43.1 Detailed Description

An imposter struct to simulate an instance.

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.44 uNature.Core.FoliageClasses.RenderingQueueReceiver Class Reference

An object that can handle an object queue.

Public Member Functions

- void **CheckPositionChange** ()
- void **ResetDensity** ()

Public Attributes

- Transform **transform**
- Camera **camera**

Properties

- [FoliageCore_Chunk](#)[] **neighbors** [get, set]

5.44.1 Detailed Description

An object that can handle an object queue.

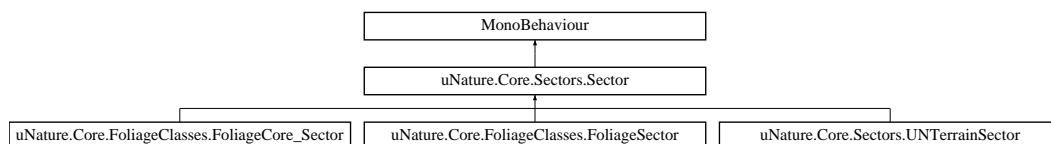
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/RenderingPipelineUtility.cs

5.45 uNature.Core.Sectors.Sector Class Reference

A sector which is used to divide the UNTerrain objects in the world to increase performance (can handle more than 200k trees!!)

Inheritance diagram for uNature.Core.Sectors.Sector:



Public Member Functions

- virtual void [OnCreated](#) (Transform owner, int resolution)
Called when the object is created.

Parameters

terrain	The terrain we belong to.
---------	---------------------------

- virtual void [Awake](#) ()
Called on awake.
- void [ResetChunks](#) ()
This method will reset the chunks' propoties, so it can be used again instead of recreating the whole sector.
- [Chunk](#) [getChunk](#) (Vector2 pos, float offset)
Get a chunk on a certain local space position
- [Chunk](#) [getChunk](#) (Vector3 pos, float offset)
Get a chunk on a certain local space position
- [Chunk](#) [getChunk](#) (Vector3 pos)
Get a chunk on a certain local space position
- List< [Chunk](#) > [getChunks](#) (Vector2 pos, float offset, bool sortResult)
Get all of the chunks that contains this specific position
- List< [Chunk](#) > [getChunks](#) (Vector3 pos, float offset, bool sortResult)
Get all of the chunks that contains this specific position
- virtual void [ApplicationQuit](#) ()
This method will be called when the application quits, used to revert all changes on terrain.

Static Public Member Functions

- static T [GenerateSector](#)< T, T1 > (Transform owner, Vector3 bounds, T sector, int res)
Generate a new sector

Public Attributes

- const int **resolutionLimit** = 40
- int **sectorResolution**
- Transform **sectorOwner**
- List< [Chunk](#) > **chunks** = new List<[Chunk](#)>()

Protected Member Functions

- virtual void [OnResolutionChanged](#) ()
Called when the resolution has been updated.
- virtual void [OnChunkCreated](#) ([Chunk](#) chunk)
Called when a chunk is created to allow custom logic on the inherited sectors.
- virtual void [OnStartCreatingChunks](#) ()
Called right before starting to create the chunks.

Properties

- Vector2 **chunkSize** [get]

Events

- SectorRecalculated **OnSectorRecalculated**

5.45.1 Detailed Description

A sector which is used to divide the UNTerrain objects in the world to increase performance (can handle more than 200k trees!!)

5.45.2 Member Function Documentation

5.45.2.1 `virtual void uNature.Core.Sectors.Sector.ApplicationQuit () [virtual]`

This method will be called when the application quits, used to revert all changes on terrain.

Reimplemented in [uNature.Core.Sectors.UNTerrainSector](#).

5.45.2.2 `virtual void uNature.Core.Sectors.Sector.Awake () [virtual]`

Called on awake.

Parameters

<i>terrain</i>	The terrain we belong to
----------------	--------------------------

Reimplemented in [uNature.Core.Sectors.UNTerrainSector](#).

5.45.2.3 `static T uNature.Core.Sectors.Sector.GenerateSector< T, T1 > (Transform owner, Vector3 bounds, T sector, int res) [static]`

Generate a new sector

Parameters

<i>terrain</i>	The terrain this sector will be generated on
<i>res</i>	the resolution of the sector (how many times will it be sliced)

Returns

The new generated sector.

Type Constraints

***T* : Sector**

***T1* : Chunk**

5.45.2.4 `Chunk uNature.Core.Sectors.Sector.getChunk (Vector2 pos, float offset)`

Get a chunk on a certain local space position

Parameters

<i>pos</i>	the local space position
<i>offset</i>	the offset (The bigger it is, the farder chunks it will find)

Returns

5.45.2.5 **Chunk** uNature.Core.Sectors.Sector.getChunk (Vector3 *pos*, float *offset*)

Get a chunk on a certain local space position

Parameters

<i>pos</i>	the local space position
<i>offset</i>	the offset (The bigger it is, the farder chunks it will find)

Returns

5.45.2.6 **Chunk** uNature.Core.Sectors.Sector.getChunk (Vector3 *pos*)

Get a chunk on a certain local space position

Parameters

<i>pos</i>	the local space position
<i>offset</i>	the offset (The bigger it is, the farder chunks it will find)

Returns

5.45.2.7 **List<Chunk>** uNature.Core.Sectors.Sector.getChunks (Vector2 *pos*, float *offset*, bool *sortResult*)

Get all of the chunks that contains this specific position

Parameters

<i>pos</i>	a local space position
<i>offset</i>	the offset (The bigger it is, the farder chunks it will find)

Returns

The chunks that contains the local space position

5.45.2.8 List<Chunk> uNature.Core.Sectors.Sector.getChunks (Vector3 *pos*, float *offset*, bool *sortResult*)

Get all of the chunks that contains this specific position

Parameters

<i>pos</i>	a local space position
<i>offset</i>	the offset (The bigger it is, the farder chunks it will find)

Returns

The chunks that contains the local space position

5.45.2.9 virtual void uNature.Core.Sectors.Sector.OnChunkCreated (Chunk *chunk*) [protected],[virtual]

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

<i>chunk</i>	
--------------	--

Reimplemented in [uNature.Core.Sectors.UNTerrainSector](#), [uNature.Core.FoliageClasses.FoliageSector](#), and [uNature.Core.FoliageClasses.FoliageCore_Sector](#).

5.45.2.10 virtual void uNature.Core.Sectors.Sector.OnCreated (Transform *owner*, int *resolution*) [virtual]

Called when the object is created.

Parameters

<i>terrain</i>	The terrain we belong to.
----------------	---------------------------

Reimplemented in [uNature.Core.Sectors.UNTerrainSector](#).

5.45.2.11 virtual void uNature.Core.Sectors.Sector.OnResolutionChanged () [protected],[virtual]

Called when the resolution has been updated.

Reimplemented in [uNature.Core.FoliageClasses.FoliageCore_Sector](#).

5.45.2.12 `virtual void uNature.Core.Sectors.Sector.OnStartCreatingChunks () [protected],[virtual]`

Called right before starting to create the chunks.

Reimplemented in [uNature.Core.Sectors.UNTerrainSector](#), [uNature.Core.FoliageClasses.FoliageSector](#), and [uNature.Core.FoliageClasses.FoliageCore_Sector](#).

5.45.2.13 `void uNature.Core.Sectors.Sector.ResetChunks ()`

This method will reset the chunks' propoties, so it can be used again instead of recreating the whole sector.

Resets:

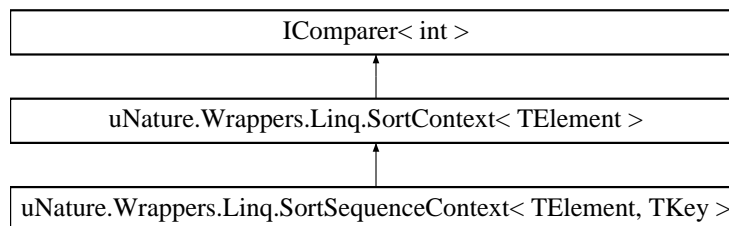
TreeInstances

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/Sector.cs

5.46 uNature.Wrappers.Linq.SortContext< TElement > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.SortContext< TElement >:



Public Member Functions

- abstract void **Initialize** (TElement[] elements)
- abstract int **Compare** (int first_index, int second_index)

Protected Member Functions

- **SortContext** (SortDirection direction, [SortContext< TElement >](#) child_context)

Protected Attributes

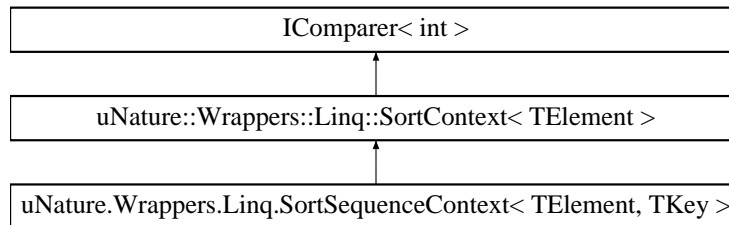
- SortDirection **direction**
- [SortContext< TElement >](#) **child_context**

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.47 uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey > Class Template Reference

Inheritance diagram for uNature.Wrappers.Linq.SortSequenceContext< TElement, TKey >:



Public Member Functions

- **SortSequenceContext** (Func< TElement, TKey > selector, IComparer< TKey > comparer, SortDirection direction, [SortContext](#)< TElement > child_context)
- override void **Initialize** (TElement[] elements)
- override int **Compare** (int first_index, int second_index)

Additional Inherited Members

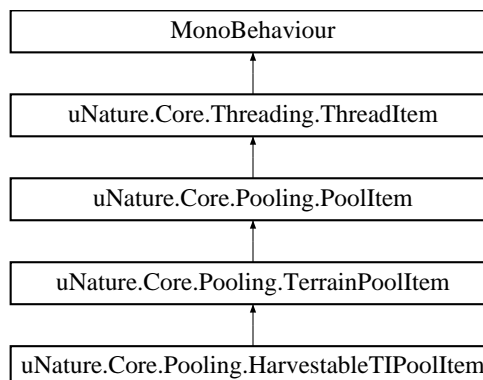
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Wrappers/LinqWrapper/LinqWrapper.cs

5.48 uNature.Core.Pooling.TerrainPoolItem Class Reference

A [Pool](#) item for terrain. (Tree instances)

Inheritance diagram for uNature.Core.Pooling.TerrainPoolItem:



Public Member Functions

- override void [MoveItem](#) (Vector3 position)
Move with rigidbody to avoid colliders movement.

Static Public Member Functions

- static void [RemoveTreeInstanceFromTerrain](#) (Terrain terrain, int treeInstanceUID)
Remove a tree instance from the terrain, Allowing you to replace it with anything else - for instance, the actual game object of the tree.
- static void [ConvertTreeInstanceOnTerrain](#) (Terrain terrain, int treeInstanceUID)
Remove a tree instance from the terrain, And replace it with a [Pool](#) item.
- static void [RestoreTreeInstanceToTerrain](#) (Terrain terrain, int treeInstanceUID)
Restore the tree instance back into the terrain.

Public Attributes

- bool [isCollider](#)
is this instance a collider ? or an actual tree instance ?
- Terrain [_terrain](#)
The terrain which owns this [Pool](#) item.

Static Public Attributes

- static bool [canModify](#) = true
Can this machine modify tree instances?
- static bool [canRestore](#) = true
Can this machine restore tree instances?

Properties

- Rigidbody **rigid** [get]
- Terrain **terrain** [get]

Events

- static OnTreeInstanceStateChanged **OnTreeInstanceRemoved**
- static OnTreeInstanceStateChanged **OnTreeInstanceRestored**

Additional Inherited Members

5.48.1 Detailed Description

A [Pool](#) item for terrain. (Tree instances)

5.48.2 Member Function Documentation

5.48.2.1 static void uNature.Core.Pooling.TerrainPoolItem.ConvertTreeInstanceOnTerrain (Terrain terrain, int treeInstanceUID)
[static]

Remove a tree instance from the terrain, And replace it with a [Pool](#) item.

5.48.2.2 override void uNature.Core.Pooling.TerrainPoolItem.MoveItem (Vector3 position) [virtual]

Move with rigidbody to avoid colliders movement.

Parameters

<i>position</i>	target position
-----------------	-----------------

Reimplemented from [uNature.Core.Pooling.PoolItem](#).

5.48.2.3 `static void uNature.Core.Pooling.TerrainPoolItem.RemoveTreeInstanceFromTerrain (Terrain terrain, int treeInstanceUID)` `[static]`

Remove a tree instance from the terrain, Allowing you to replace it with anything else - for instance, the actual game object of the tree.

5.48.2.4 `static void uNature.Core.Pooling.TerrainPoolItem.RestoreTreeInstanceToTerrain (Terrain terrain, int treeInstanceUID)` `[static]`

Restore the tree instance back into the terrain.

5.48.3 Member Data Documentation

5.48.3.1 `Terrain uNature.Core.Pooling.TerrainPoolItem._terrain`

The terrain which owns this [Pool](#) item.

5.48.3.2 `bool uNature.Core.Pooling.TerrainPoolItem.canModify = true` `[static]`

Can this machine modify tree instances?

5.48.3.3 `bool uNature.Core.Pooling.TerrainPoolItem.canRestore = true` `[static]`

Can this machine restore tree instances?

5.48.3.4 `bool uNature.Core.Pooling.TerrainPoolItem.isCollider`

is this instance a collider ? or an actual tree instance ?

The documentation for this class was generated from the following file:

- `D:/Projects/uNature/Assets/uNature/Scripts/Core/Pool/PoolItems/TerrainPoolItem.cs`

5.49.2 Member Function Documentation

5.49.2.1 `virtual void uNature.Core.Threading.ThreadItem.OnDisable () [protected], [virtual]`

Called when the object is disabled

Reimplemented in [uNature.Core.Terrains.UNTerrain](#), [uNature.Core.FoliageClasses.FoliageManagerInstance](#), [uNature.Core.FoliageClasses.FoliageCore_MainManager](#), [uNature.Core.FoliageClasses.FoliageMeshManager](#), [uNature.Core.FoliageClasses.TouchBending](#), [uNature.Core.Targets.UNTarget](#), [uNature.Core.FoliageClasses.FoliageReceiver](#), [uNature.Core.Pooling.PoolItem](#), and [uNature.Core.FoliageClasses.BaseInteraction](#).

5.49.2.2 `virtual void uNature.Core.Threading.ThreadItem.OnPositionChanged (Vector3 newPosition) [protected], [virtual]`

Called when the item's position changed

Reimplemented in [uNature.Core.Terrains.UNTerrain](#), [uNature.Core.FoliageClasses.TouchBending](#), and [uNature.Core.FoliageClasses.BaseInteraction](#).

5.49.2.3 `virtual void uNature.Core.Threading.ThreadItem.Update () [protected], [virtual]`

Update...

Reimplemented in [uNature.Core.FoliageClasses.FoliageMeshManager](#), [uNature.Core.FoliageClasses.FoliageCore_MainManager](#), [uNature.Core.FoliageClasses.FoliageReceiver](#), [uNature.Core.Targets.UNTarget](#), and [uNature.Core.Seekers.UNSeeker](#).

5.49.2.4 `virtual void uNature.Core.Threading.ThreadItem.UpdateItem () [virtual]`

This method will update this thread item, called externally from unity's main thread.

5.49.3 Member Data Documentation

5.49.3.1 `List<ThreadItem> uNature.Core.Threading.ThreadItem._threadItems [static]`

A list that holds all of the thread items in the scene.

The documentation for this class was generated from the following file:

- `D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/ThreadItem.cs`

5.50 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.50.1 Detailed Description

A thread task that takes no parameters.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.51 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.51.1 Detailed Description

A thread task that takes no parameters.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.52 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.52.1 Detailed Description

A thread task that takes no parameters.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.53 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.53.1 Detailed Description

A thread task that takes no parameters.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.54 uNature.Core.Threading.ThreadTask Class Reference

A thread task that takes no parameters.

5.54.1 Detailed Description

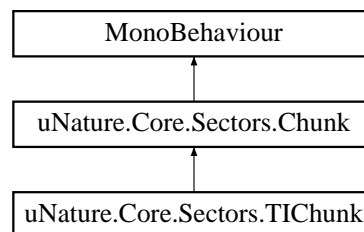
A thread task that takes no parameters.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.55 uNature.Core.Sectors.TIChunk Class Reference

Inheritance diagram for uNature.Core.Sectors.TIChunk:



Public Member Functions

- override void [Awake](#) ()
Called on awake
- override void [OnDrawGizmos](#) ()
Draw gizmos.
- override void [OnCreated](#) ()
Called when created.
- virtual void [GenerateTreeInstances](#) (TreeInstance[] trees, Vector3 terrainSize, TerrainData tData, Vector3 terrainPos)
Generate tree instances, derived from a certain provided tree instances
- void [AddTreeInstance](#) (int instanceID, Vector3 terrainSize, TreeInstance treeInstance, TerrainData terrainData, Vector3 terrainPos, [UNTerrainSector](#) sector)
Add a tree instance into this chunk
- override void [ResetChunk](#) ()
Reset chunk.
- void [CheckForNearbyTreeInstances](#) ([UNSeeker](#) seeker, [UNTerrain](#) terrain)
Check and assign nearby tree instances.

Public Attributes

- List< int > **objectsInstanceIDs** = new List<int>()
- List< [ChunkObject](#) > **objects** = new List<[ChunkObject](#)>()

Protected Member Functions

- override void [OnSizeChanged](#) ()
Called when the size is changed

Properties

- override string **chunkType** [get]
- Terrain **terrain** [get, set]

Additional Inherited Members

5.55.1 Member Function Documentation

5.55.1.1 void [uNature.Core.Sectors.TIChunk.AddTreeInstance](#) (int *instanceID*, Vector3 *terrainSize*, TreeInstance *treeInstance*, TerrainData *terrainData*, Vector3 *terrainPos*, UNTerrainSector *sector*)

Add a tree instance into this chunk

Parameters

<i>instanceID</i>	the targeted tree instance.
<i>treeInstance</i>	the tree instance you want to add
<i>terrainData</i>	the terrain data that this chunk belongs to

5.55.1.2 override void [uNature.Core.Sectors.TIChunk.Awake](#) () [virtual]

Called on awake

Parameters

<i>terrain</i>	
<i>terrainBase</i>	

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.55.1.3 void [uNature.Core.Sectors.TIChunk.CheckForNearbyTreeInstances](#) (UNSeeker *seeker*, UNTerrain *terrain*)

Check and assign nearby tree instances.

5.55.1.4 virtual void [uNature.Core.Sectors.TIChunk.GenerateTreeInstances](#) (TreeInstance[] *trees*, Vector3 *terrainSize*, TerrainData *tData*, Vector3 *terrainPos*) [virtual]

Generate tree instances, derived from a certain provided tree instances

Parameters

<i>trees</i>	the tree instances
<i>tData</i>	the terrain data

5.55.1.5 override void uNature.Core.Sectors.TIChunk.OnCreated () [virtual]

Called when created.

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.55.1.6 override void uNature.Core.Sectors.TIChunk.OnDrawGizmos () [virtual]

Draw gizmos.

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.55.1.7 override void uNature.Core.Sectors.TIChunk.OnSizeChanged () [protected], [virtual]

Called when the size is changed

Reimplemented from [uNature.Core.Sectors.Chunk](#).

5.55.1.8 override void uNature.Core.Sectors.TIChunk.ResetChunk () [virtual]

Reset chunk.

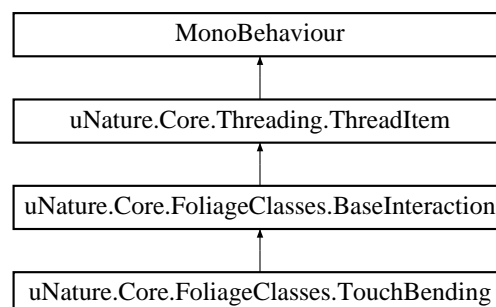
Reimplemented from [uNature.Core.Sectors.Chunk](#).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Sectors/TIChunk.cs

5.56 uNature.Core.FoliageClasses.TouchBending Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.TouchBending:



Public Attributes

- bool **inBounds** = false

Static Public Attributes

- static Vector4[] **bendingTargets** = new Vector4[20]

Protected Member Functions

- override void **OnEnable** ()
- override void **OnDisable** ()
Called when the object is disabled
- virtual void **OnDrawGizmos** ()
- override void **OnPositionChanged** (Vector3 newPosition)
Called when the item's position changed

Properties

- bool **simulateOnEditorTime** [get, set]
- float **radius** [get, set]
- float **seekingRange** [get, set]
- int **id** [get]
- bool **simulate** [get]
- override bool **includedInInteractionMap** [get]

Additional Inherited Members

5.56.1 Member Function Documentation

5.56.1.1 override void uNature.Core.FoliageClasses.TouchBending.OnDisable () [protected],[virtual]

Called when the object is disabled

Reimplemented from [uNature.Core.FoliageClasses.BaseInteraction](#).

5.56.1.2 override void uNature.Core.FoliageClasses.TouchBending.OnPositionChanged (Vector3 newPosition)
[protected],[virtual]

Called when the item's position changed

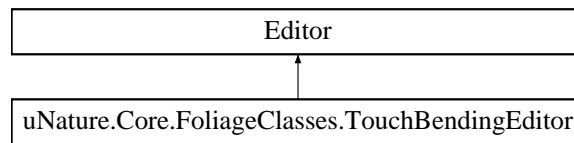
Reimplemented from [uNature.Core.FoliageClasses.BaseInteraction](#).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/Interaction/TouchBending.cs

5.57 uNature.Core.FoliageClasses.TouchBendingEditor Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.TouchBendingEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/TouchBendingEditor.cs

5.58 uNature.Core.Sectors.TreeFetchingTask_MultiThreaded Struct Reference

Public Member Functions

- **TreeFetchingTask_MultiThreaded** (TreeInstance[] treeInstances, TreePrototype[] treePrototypes, TerrainData tData, bool isRunning, System.Action OnFinish)

Public Attributes

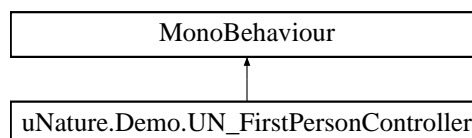
- TreeInstance[] **treeInstances**
- TreePrototype[] **treePrototypes**
- TerrainData **tData**
- bool **isRunning**
- System.Action **OnFinish**

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/UNTerrainSector.cs

5.59 uNature.Demo.UN_FirstPersonController Class Reference

Inheritance diagram for uNature.Demo.UN_FirstPersonController:



Public Attributes

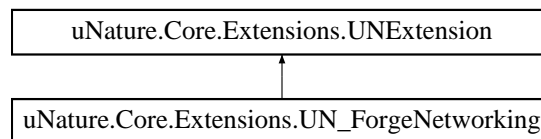
- bool **getInputsMouse** = true

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Demo/UN_FirstPersonController.cs

5.60 uNature.Core.Extensions.UN_ForgeNetworking Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_ForgeNetworking:



Public Member Functions

- void **CreateManager** ()

Properties

- override string **AssetName** [get]
- override string **AssetDescription** [get]
- override string **AssetNameSpace** [get]
- override string **AssetStoreAddress** [get]
- override string **PublisherName** [get]
- override string **AssetLogoName** [get]
- override string **AssetDocumentationName** [get]

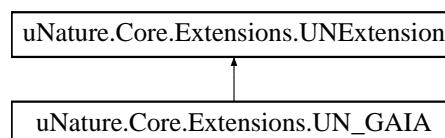
Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/ForgeNetworking/UN_ForgeNetworking.cs

5.61 uNature.Core.Extensions.UN_GAIA Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_GAIA:



Properties

- override string **AssetName** [get]
- override string **AssetDescription** [get]
- override string **AssetNameSpace** [get]
- override bool **IsDefault** [get]
- override string **AssetStoreAddress** [get]
- override string **PublisherName** [get]
- override string **AssetLogoName** [get]
- override string **AssetDocumentationName** [get]

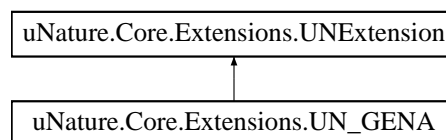
Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/GAIA/UN_GAIA.cs

5.62 uNature.Core.Extensions.UN_GENA Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_GENA:



Properties

- override string **AssetName** [get]
- override string **AssetDescription** [get]
- override string **AssetNameSpace** [get]
- override bool **IsDefault** [get]
- override string **AssetStoreAddress** [get]
- override string **PublisherName** [get]
- override string **AssetLogoName** [get]
- override string **AssetDocumentationName** [get]

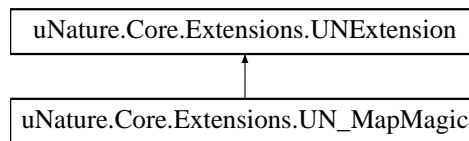
Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/GENA/UN_GENA.cs

5.63 uNature.Core.Extensions.UN_MapMagic Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_MapMagic:



Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override bool **Featured** [get]
- override string **AssetDocumentationName** [get]
- override string **AssetDescription** [get]
- override string **AssetStoreAdress** [get]
- override string **AssetLogoName** [get]

Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/MapMagic/UN_MapMagic.cs

5.64 uNature.Demo.UN_MouseLook Class Reference

Public Member Functions

- void **Init** (Transform character, Transform camera)
- void **LookRotation** (Transform character, Transform camera)

Public Attributes

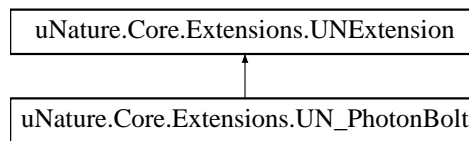
- float **XSensitivity** = 2f
- float **YSensitivity** = 2f
- bool **clampVerticalRotation** = true
- float **MinimumX** = -90F
- float **MaximumX** = 90F
- bool **smooth**
- float **smoothTime** = 5f

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Demo/UN_MouseLook.cs

5.65 uNature.Core.Extensions.UN_PhotonBolt Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_PhotonBolt:



Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override string **AssetDocumentationName** [get]
- override string **AssetDescription** [get]
- override string **AssetStoreAdress** [get]
- override string **AssetLogoName** [get]

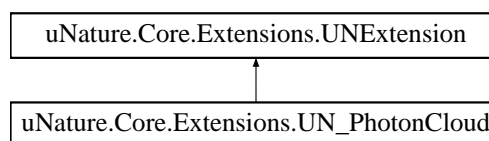
Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/PhotonBolt/UN_PhotonBolt.cs

5.66 uNature.Core.Extensions.UN_PhotonCloud Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_PhotonCloud:



Public Member Functions

- void **CreateManager** ()

Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override string **AssetDocumentationName** [get]
- override string **AssetDescription** [get]
- override string **AssetStoreAdress** [get]
- override string **AssetLogoName** [get]

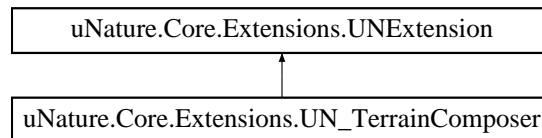
Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/PhotonCloud/UN_PhotonCloud.cs

5.67 uNature.Core.Extensions.UN_TerrainComposer Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_TerrainComposer:



Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override bool **Featured** [get]
- override bool **IsDefault** [get]
- override string **AssetDescription** [get]
- override string **AssetStoreAddress** [get]
- override string **AssetLogoName** [get]

Additional Inherited Members

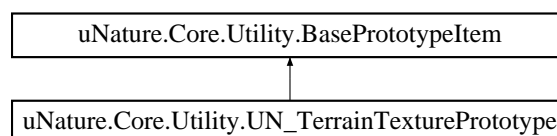
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/TC2/UN_TerrainComposer.cs

5.68 uNature.Core.Utility.UN_TerrainTexturePrototype Class Reference

The prototype of the terrain texture. [Used for listing the paintable surfaces]

Inheritance diagram for uNature.Core.Utility.UN_TerrainTexturePrototype:



Public Member Functions

- **UN_TerrainTexturePrototype** (Texture2D splatTexture)

Public Attributes

- Texture2D **splatTexture**

Protected Member Functions

- override Texture2D **GetPreview** ()
Get a preview of the splat texture.

Additional Inherited Members

5.68.1 Detailed Description

The prototype of the terrain texture. [Used for listing the paintable surfaces]

5.68.2 Member Function Documentation

5.68.2.1 override Texture2D uNature.Core.Utility.UN_TerrainTexturePrototype.GetPreview () [protected],
[virtual]

Get a preview of the splat texture.

Returns

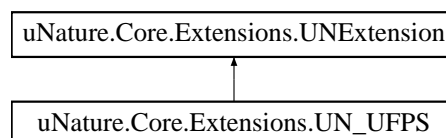
Reimplemented from [uNature.Core.Utility.BasePrototypeItem](#).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNBrushUtility.cs

5.69 uNature.Core.Extensions.UN_UFPS Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_UFPS:



Public Member Functions

- void **ApplyOnCurrentPool** ()

Properties

- override string **AssetName** [get]
- override string **AssetDescription** [get]
- override string **AssetLogoName** [get]
- override string **AssetNameSpace** [get]
- override string **AssetStoreAddress** [get]
- override string **PublisherName** [get]
- override string **AssetDocumentationName** [get]

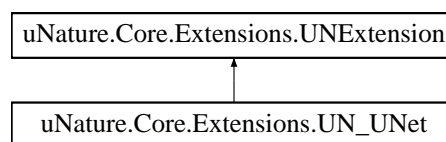
Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/UFPS/UN_UFPS.cs

5.70 uNature.Core.Extensions.UN_UNet Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_UNet:



Public Member Functions

- void **CreateManager** ()

Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override string **AssetDocumentationName** [get]

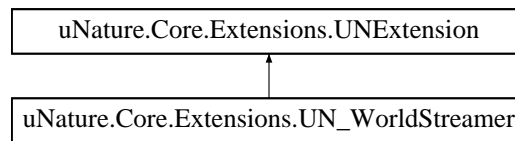
Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/UNet/UN_UNet.cs

5.71 uNature.Core.Extensions.UN_WorldStreamer Class Reference

Inheritance diagram for uNature.Core.Extensions.UN_WorldStreamer:



Properties

- override string **AssetName** [get]
- override string **AssetNameSpace** [get]
- override string **PublisherName** [get]
- override bool **Featured** [get]
- override string **AssetDocumentationName** [get]
- override string **AssetDescription** [get]
- override string **AssetStoreAddress** [get]
- override string **AssetLogoName** [get]

Additional Inherited Members

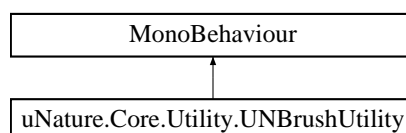
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Integrations/WorldStreamer/UN_WorldStreamer.cs

5.72 uNature.Core.Utility.UNBrushUtility Class Reference

Using this class you can paint an brush on the scene.

Inheritance diagram for uNature.Core.Utility.UNBrushUtility:



Public Member Functions

- void **DrawBrush** (Texture2D brushTexture, Color brushColor, Vector3 originPosition, Quaternion originRotation, float brushSize)
Draw a brush on the scene.

Static Public Member Functions

- static Texture2D [Resize](#) (Texture2D source, int newWidth, int newHeight)
Resize texture by Justin Markwell and Smoke.
- static float [CheckSplatPaint](#) (RaycastHit hit, Vector3 worldPosition, List< [UN_TerrainTexturePrototype](#) > chosenSplats)
Checks if the splats are in that specific position.

Properties

- static [UNBrushUtility](#) **instance** [get]
- static Projector **projector** [get]
- List< [UN_TerrainTexturePrototype](#) > **splatPrototypes** [get]

5.72.1 Detailed Description

Using this class you can paint an brush on the scene.

5.72.2 Member Function Documentation

- 5.72.2.1 static float uNature.Core.Utility.UNBrushUtility.CheckSplatPaint (RaycastHit *hit*, Vector3 *worldPosition*, List< [UN_TerrainTexturePrototype](#) > *chosenSplats*) [static]

Checks if the splats are in that specific position.

Parameters

<i>hit</i>	
<i>worldPosition</i>	

Returns

- 5.72.2.2 void uNature.Core.Utility.UNBrushUtility.DrawBrush (Texture2D *brushTexture*, Color *brushColor*, Vector3 *originPosition*, Quaternion *originRotation*, float *brushSize*)

Draw a brush on the scene.

Parameters

<i>brushTexture</i>	The brush's texture.
<i>brushColor</i>	The brush's color.
<i>position</i>	The brush's origin position (for example the camera's position).
<i>rotation</i>	The brush's origin rotation (for example the camera's rotation).
<i>brushSize</i>	The brush's size. (Varies from 1 -> 100)

5.72.2.3 static Texture2D uNature.Core.Utility.UNBrushUtility.Resize (Texture2D *source*, int *newWidth*, int *newHeight*)
[static]

Resize texture by Justin Markwell and Smoke.

Parameters

<i>source</i>	
<i>newWidth</i>	
<i>newHeight</i>	

Returns

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNBrushUtility.cs

5.73 uNature.Core.Utility.UNCombineInstance Struct Reference

Public Member Functions

- **UNCombineInstance** (Matrix4x4 transform, Mesh mesh, float spread, int density, int id)

Public Attributes

- Matrix4x4 **transform**
- Mesh **mesh**
- Vector2 **densityOffset**
- int **density**

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNBatchUtility.cs

5.74 uNature.Core.Utility.UNDictionary< T, T1 > Class Template Reference

Public Member Functions

- void **Add** (T key, T1 value)
- void **RemoveAt** (int index)
- void **Remove** (T key)
- int **TryGetKeyIndex** (T key)

Properties

- List< T > **Keys** [get]
- List< T1 > **Values** [get]
- int **Count** [get]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNDictionary.cs

5.75 uNature.Core.Collections.UNDimensionalList< T > Class Template Reference

A 2 dimensional list which is used by certain mechanics in [uNature](#).

Public Member Functions

- bool [ContainsKey](#) (int key)
Checks if the list contains a certain key.
- bool [ContainsValue](#) (T value)
Does the two dimensional list contain this value?
- void [TryAddKey](#) (List< T > value)
Try to add a key.

Properties

- List< T > [this\[int index\]](#) [get, set]
Get the stashed list.
- int [Count](#) [get]
Count of the two dimensional list elements.

5.75.1 Detailed Description

A 2 dimensional list which is used by certain mechanics in [uNature](#).

5.75.2 Member Function Documentation

5.75.2.1 bool uNature.Core.Collections.UNDimensionalList< T >.ContainsKey (int key)

Checks if the list contains a certain key.

5.75.2.2 bool uNature.Core.Collections.UNDimensionalList< T >.ContainsValue (T value)

Does the two dimensional list contain this value?

Parameters

<i>value</i>	the value
--------------	-----------

Returns

is it contained ?

5.75.2.3 void `uNature.Core.Collections.UNDimensionalList< T >.TryAddKey (List< T > value)`

Try to add a key.

Parameters

<i>value</i>	the value
--------------	-----------

5.75.3 Property Documentation

5.75.3.1 int `uNature.Core.Collections.UNDimensionalList< T >.Count` [get]

Count of the two dimensional list elements.

5.75.3.2 List<T> `uNature.Core.Collections.UNDimensionalList< T >.this[int index]` [get], [set]

Get the stashed list.

Parameters

<i>index</i>	index
--------------	-------

Returns

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Collections/UNDimensionalList.cs

5.76 uNature.Core.Editor.Helpers.UNEditorHelpers Class Reference

Static Public Member Functions

- static void **SetupSceneTerrains** ()

- static void **FixCorruptedTreeInstanceOnSceneTerrains** ()
- static void **CopySelectedTerrains** ()
- static void **UpdateSelectedTerrains** ()
- static void **ShowDebugWindow** ()

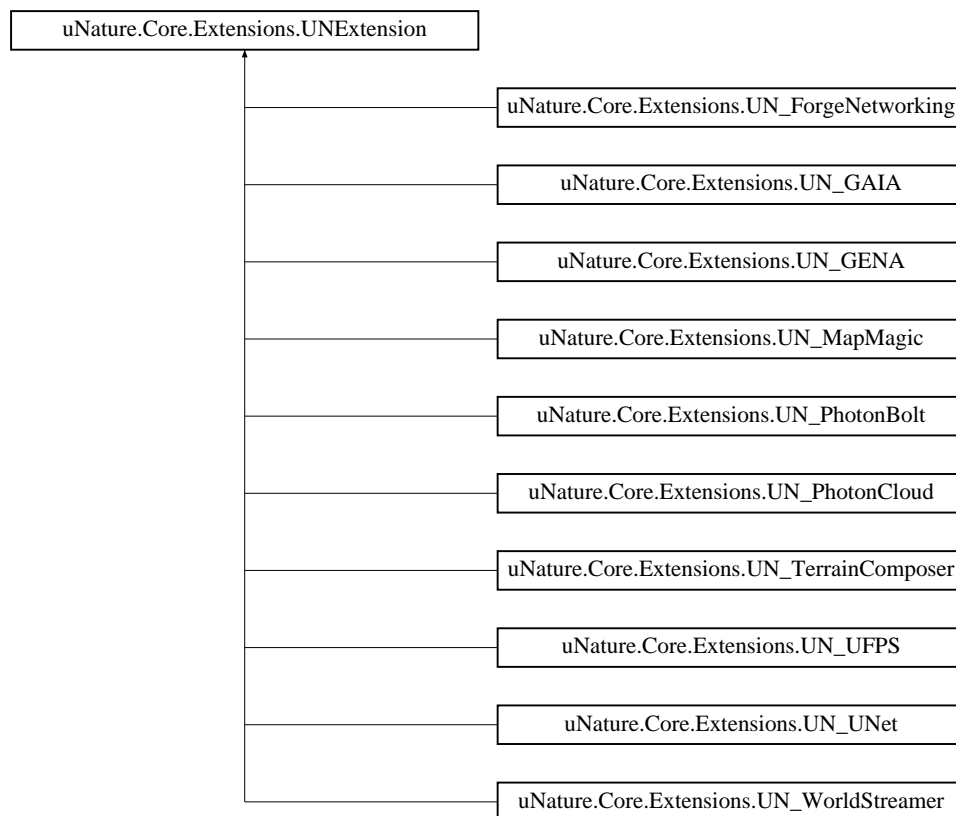
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNEditorHelpers.cs

5.77 uNature.Core.Extensions.UNExtension Class Reference

A uConstruct extension that will allow other 3d party systems to work with uConstruct.

Inheritance diagram for uNature.Core.Extensions.UNExtension:



Static Public Member Functions

- static void [OpenDocs](#) ([UNExtension](#) instance)
Open the documentation of the extension

Parameters

instance	<i>Extension instance</i>
----------	---------------------------

- static void [OpenAssetStore](#) ([UNExtension](#) instance)

Open the asset store page of the extension

Parameters

instance	<i>Extension instance</i>
----------	---------------------------

- static Texture [GetLogo](#) ([UNExtension](#) instance)
Get the extension logo
- static void [LoadMethods](#) ([UNExtension](#) instance, Type type)
Load helper methods from an instance.

Public Attributes

- bool [isViewed](#)
Is the extension viewed?
- List< MethodInfo > [HelperMethods](#)
Loaded methods that are created to give tools to people who activated the extension.

Properties

- virtual string [AssetName](#) [get]
The asset name (for example TreesManagerSystem).
- virtual string [AssetDescription](#) [get]
The asset description (for example :
- virtual bool [Featured](#) [get]
Is this asset featured?
- virtual string [AssetLogoName](#) [get]
Asset logo name that will be searched on the project.
- virtual string [PublisherName](#) [get]
The asset publisher name (for example EEProductions).
- virtual string [AssetDocumentationName](#) [get]
Asset extension documentation name.
- virtual string [AssetStoreAdress](#) [get]
Asset extension asset store adress - (For exmaple - <https://www.assetstore.unity3d.com/en/#!/content/43129>).
- virtual string [AssetNameSpace](#) [get]
The namespace that will be added to the defines when the extension is activated.
- virtual bool [IsDefault](#) [get]
Default means that this asset doesnt require it to be enabled, that means that its working with uConstruct out of the box.
- bool [isActivated](#) [get, set]
Is the extension activated currently?

5.77.1 Detailed Description

A uConstruct extension that will allow other 3d party systems to work with uConstruct.

5.77.2 Member Function Documentation

5.77.2.1 static Texture uNature.Core.Extensions.UNExtension.GetLogo ([UNExtension instance](#)) [static]

Get the extension logo

Parameters

<i>instance</i>	Extension instance
-----------------	--------------------

5.77.2.2 `static void uNature.Core.Extensions.UNExtension.LoadMethods (UNExtension instance, Type type)`
`[static]`

Load helper methods from an instance.

Parameters

<i>instance</i>	Extension instance
-----------------	--------------------

5.77.2.3 `static void uNature.Core.Extensions.UNExtension.OpenAssetStore (UNExtension instance)` `[static]`

Open the asset store page of the extension

Parameters

<i>instance</i>	Extension instance
-----------------	--------------------

5.77.2.4 `static void uNature.Core.Extensions.UNExtension.OpenDocs (UNExtension instance)` `[static]`

Open the documentation of the extension

Parameters

<i>instance</i>	Extension instance
-----------------	--------------------

5.77.3 Member Data Documentation

5.77.3.1 `List<MethodInfo> uNature.Core.Extensions.UNExtension.HelperMethods`

Loaded methods that are created to give tools to people who activated the extension.

5.77.3.2 `bool uNature.Core.Extensions.UNExtension.isViewed`

Is the extension viewed?

5.77.4 Property Documentation

5.77.4.1 `virtual string uNature.Core.Extensions.UNExtension.AssetDescription` `[get]`

The asset description (for example :

An asset used for optimizing terrain & game world. Features :

5.77.4.2 virtual string uNature.Core.Extensions.UNExtension.AssetDocumentationName [get]

Asset extension documentation name.

5.77.4.3 virtual string uNature.Core.Extensions.UNExtension.AssetLogoName [get]

Asset logo name that will be searched on the project.

For example: uConstructLogo

5.77.4.4 virtual string uNature.Core.Extensions.UNExtension.AssetName [get]

The asset name (for example TreesManagerSystem).

5.77.4.5 virtual string uNature.Core.Extensions.UNExtension.AssetNameSpace [get]

The namespace that will be added to the defines when the extension is activated.

5.77.4.6 virtual string uNature.Core.Extensions.UNExtension.AssetStoreAdress [get]

Asset extension asset store adress - (For exmaple - <https://www.assetstore.unity3d.com/en/#!/content/43129>).

5.77.4.7 virtual bool uNature.Core.Extensions.UNExtension.Featued [get]

Is this asset featured?

5.77.4.8 bool uNature.Core.Extensions.UNExtension.isActivated [get], [set]

Is the extension activated currently?

5.77.4.9 virtual bool uNature.Core.Extensions.UNExtension.IsDefault [get]

Default means that this asset doesnt require it to be enabled, that means that its working with uConstruct out of the box.

5.77.4.10 virtual string uNature.Core.Extensions.UNExtension.PublisherName [get]

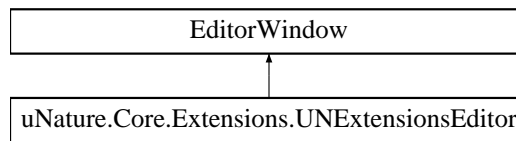
The asset publisher name (for example EEProductions).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Base/UNExtension.cs

5.78 uNature.Core.Extensions.UNExtensionsEditor Class Reference

Inheritance diagram for uNature.Core.Extensions.UNExtensionsEditor:



Static Public Member Functions

- static void **Open** ()
- static void **HandleCompile** ()

Public Attributes

- const string **UN_DEFINE** = "UN_INSTALLED"

Properties

- static Texture2D **featuredIcon** [get]
- static GUIStyle **featuredFoldoutStyle** [get]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Extensions/Editor/UNExtensionsEditor.cs

5.79 uNature.Core.Utility.UNFastList< T > Class Template Reference

A list that requires you to set

Public Member Functions

- **UNFastList** (int maxCapacity)
- void **Add** (T item)
- void **Clear** ()

Properties

- int **Count** [get]

5.79.1 Detailed Description

A list that requires you to set

Template Parameters

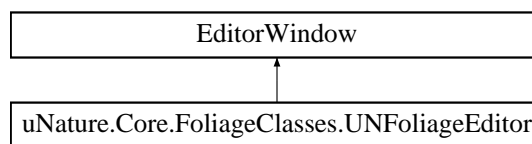
<i>T</i>	
----------	--

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNFastList.cs

5.80 uNature.Core.FoliageClasses.UNFoliageEditor Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.UNFoliageEditor:



Public Member Functions

- void **OnGUI** ()

Static Public Member Functions

- static void **OpenWindow** ()

Properties

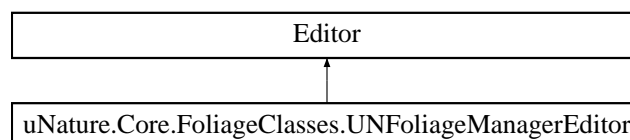
- [FoliagePrototype](#) **currentPrototype** [get, set]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNFoliageEditor.cs

5.81 uNature.Core.FoliageClasses.UNFoliageManagerEditor Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.UNFoliageManagerEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNFoliageManagerEditor.cs

5.82 uNature.Core.Collections.UNList< T > Class Template Reference

A custom list which is used on some important interfaces in UN.

Public Member Functions

- void **Add** (T item)
Add an item to the list.
- void **Remove** (T item)
Remove an item from the list.
- T **TryGet** (System.Object similarItem)
Get a similar instance by a custom Equals which needs to be initialized on the item.
- bool **Contains** (System.Object item)
Is this item contained in the list?

Properties

- int **Count** [get]
Get list count
- T **this[int index]** [get, set]
Get an element from the list.

5.82.1 Detailed Description

A custom list which is used on some important interfaces in UN.

Template Parameters

<i>T</i>	the list type.
----------	----------------

5.82.2 Member Function Documentation

5.82.2.1 void uNature.Core.Collections.UNList< T >.Add (T item)

Add an item to the list.

Parameters

<i>item</i>	
-------------	--

5.82.2.2 bool uNature.Core.Collections.UNList< T >.Contains (System.Object *item*)

Is this item contained in the list?

Parameters

<i>item</i>	
-------------	--

Returns

5.82.2.3 void uNature.Core.Collections.UNList< T >.Remove (T *item*)

Remove an item from the list.

Parameters

<i>item</i>	
-------------	--

5.82.2.4 T uNature.Core.Collections.UNList< T >.TryGet (System.Object *similarItem*)

Get a similar instance by a custom Equals which needs to be initialized on the item.

Parameters

<i>similarItem</i>	
--------------------	--

Returns

5.82.3 Property Documentation**5.82.3.1 int uNature.Core.Collections.UNList< T >.Count [get]**

Get list count

5.82.3.2 T uNature.Core.Collections.UNList< T >.this[int index] [get], [set]

Get an element from the list.

Parameters

index	
-----------------------	--

Returns

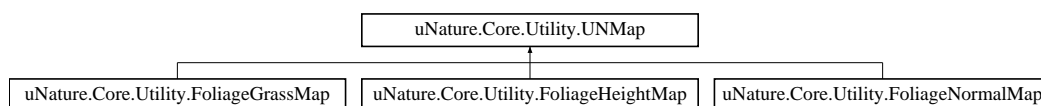
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Collections/UNList.cs

5.83 uNature.Core.Utility.UNMap Class Reference

The abstract Map class.

Inheritance diagram for uNature.Core.Utility.UNMap:



Public Member Functions

- void **RestoreChanges** ()
- void **Apply** (Color32[] pixels)
- void **SetPixels32** ()
- void **SetPixels32** (Color32[] pixels)
- void **SetPixelsNoApply** ()
- void **SetPixels32Delayed** ()
- byte[] **EncodeToPNG** ()
- void **Resize** (int size)
- void **Clear** (bool autoApply, Color32 defaultColor)

Protected Member Functions

- **UNMap** (Texture2D texture, Color32[] pixels, [FoliageManagerInstance](#) mInstance)
- virtual void **OnDirty** (bool value)

Protected Attributes

- Texture2D **_map**
- Color32[] **originalMapPixels** = null

Properties

- Texture2D **map** [get, set]
- Color32[] **mapPixels** [get, set]
- int **mapWidth** [get]
- bool **dirty** [get, set]
- [FoliageManagerInstance](#) **mInstance** [get]

5.83.1 Detailed Description

The abstract Map class.

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMapGenerators.cs

5.84 uNature.Core.FoliageClasses.UNMeshData Class Reference

Public Member Functions

- **UNMeshData** (Mesh mesh)

Public Attributes

- List< Vector3 > **vertices**
- List< Vector3 > **normals**
- List< int > **triangles**
- List< Vector2 > **uv1s**
- int **verticesLength**
- int **normalsLength**
- int **uv1sLength**
- int **trianglesLength**

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliagePrototype.cs

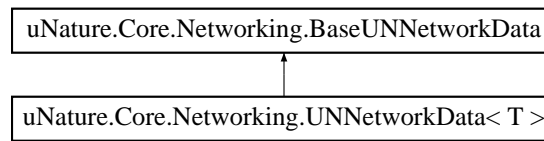
5.85 uNature.Core.Networking.UNNetworkData< T > Class Template Reference

A class which can be used for an abstract networking data.

Template Parameters

<i>T</i>	The network connection type which the networking library uses.
----------	--

Inheritance diagram for uNature.Core.Networking.UNNetworkData< T >:



Public Member Functions

- override void [UnPack](#) ()
Unpack the data
- virtual void [SendToServer](#) ()
Send data to server
- virtual void [SendToConnection](#) (T connection)
Send data to connection

Parameters

connection	<i>the targeted connection</i>
------------	--------------------------------

- virtual void [SendToClients](#) ()
Send data to clients
- virtual void [SendToOthers](#) ()
Send data to other connections
- override bool [Equals](#) (object obj)
Create equal state which checks whether those 2 instances of NetworkData are equal
- override int [GetHashCode](#) ()
Overriden this method only to get rid of a warning.
- virtual byte[] [Serialize](#) ()
Serialize the object

Static Public Member Functions

- static T2 [Pack](#)< T1, T2 > (Terrain terrain, int treeInstanceID, int health, PacketType type)
Pack the data and create a data instance
- static [UNNetworkData](#)< T > [Deserialize](#) (byte[] bytes)
Deserialize the data.

Additional Inherited Members

5.85.1 Detailed Description

A class which can be used for an abstract networking data.

Template Parameters

<i>T</i>	The network connection type which the networking library uses.
----------	--

5.85.2 Member Function Documentation

5.85.2.1 static UNNetworkData<T> uNature.Core.Networking.UNNetworkData< T >.Deserialize (byte[] bytes)
[static]

Deserialize the data.

Parameters

<i>bytes</i>	the data.
--------------	-----------

Returns

the deserialized object.

5.85.2.2 override bool uNature.Core.Networking.UNNetworkData< T >.Equals (object obj)

Create equal state which checks whether those 2 instances of NetworkData are equal

Parameters

<i>obj</i>	
------------	--

Returns

5.85.2.3 override int uNature.Core.Networking.UNNetworkData< T >.GetHashCode ()

Overrided this method only to get rid of a warning.

Returns

5.85.2.4 static T2 uNature.Core.Networking.UNNetworkData< T >.Pack< T1, T2 > (Terrain terrain, int treeInstanceId, int health, PacketType type) [static]

Pack the data and create a data instance

Type Constraints

T2 : UNNetworkData< T1 >

5.85.2.5 `virtual void uNature.Core.Networking.UNNetworkData< T >.SendToClients ()` [virtual]

Send data to clients

5.85.2.6 `virtual void uNature.Core.Networking.UNNetworkData< T >.SendToConnection (T connection)`
[virtual]

Send data to connection

Parameters

<i>connection</i>	the targeted connection
-------------------	-------------------------

5.85.2.7 virtual void **uNature.Core.Networking.UNNetworkData**< T >.SendToOthers () [virtual]

Send data to other connections

5.85.2.8 virtual void **uNature.Core.Networking.UNNetworkData**< T >.SendToServer () [virtual]

Send data to server

5.85.2.9 virtual byte [] **uNature.Core.Networking.UNNetworkData**< T >.Serialize () [virtual]

Serialize the object

Returns

serialized bytes

5.85.2.10 override void **uNature.Core.Networking.UNNetworkData**< T >.UnPack () [virtual]

Unpack the data

Reimplemented from [uNature.Core.Networking.BaseUNNetworkData](#).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/UNNetworkData.cs

5.86 uNature.Core.Networking.UNNetworkManager< T1, T2 > Class Template Reference

A template for networking, which can be used by networking extensions to easily get the networking actions done.

Template Parameters

<i>T1</i>	the targeted networking connection
<i>T2</i>	the type of the data

Public Member Functions

- [UNNetworkManager](#) (MonoBehaviour managerInstance)

The constructor of this class, initiate basic events.

- virtual void [Awake](#) ()
Called when the network manager is initialized.
- virtual void [UpdatePermissions](#) ()
Reupdate the permissions for the Pool items. (needs to be called when ever there's a networking change for the owner/controller)
- virtual void [SendEvent](#) ([UNNetworkData](#)< T1 > instance)
Send event to the correct location
- virtual void [SendToConnection](#) (T1 connection, [UNNetworkData](#)< T1 > instance)
Send to certain connection the data
- virtual void [SendToClients](#) ([UNNetworkData](#)< T1 > instance)
Send to all of the clients.
- virtual void [SendToOthers](#) ([UNNetworkData](#)< T1 > instance)
Send to all other connections.
- virtual void [SendToServer](#) ([UNNetworkData](#)< T1 > instance)
Send to the server the data.
- void [OnClientConnected](#) (T1 conn)
Called when the client connects, send all data.

Public Attributes

- const float **STREAM_UPDATE_CHECK_INTERVAL_SECONDS** = 2

Static Public Attributes

- static [UNNetworkManager](#)< T1, T2 > [manager](#)
a static instance of this object.

Protected Member Functions

- IEnumerator [CheckForStreamingBufferedUpdates](#) ()
This method is checking every certain amount of seconds for new loaded streamed areas to update data that is waiting for a streamed terrain to be loaded.
- void [OnHarvestableTreeInstancePooled](#) ([HarvestableTIPoolItem](#) instance)
Called when an harvestable item instance has been created
- void [OnItemDamaged](#) ([HarvestableTIPoolItem](#) item, int damage)
Update item damage and handle synchorization.

Properties

- static [UNList](#)< [BaseUNNetworkData](#) > [bufferedData](#) [get]
The buffered data which will be sent to all of the connecting connections.
- virtual bool [isServer](#) [get]
Are we the server?
- virtual bool [isAuth](#) [get]
is the server architecture is authoritative?

5.86.1 Detailed Description

A template for networking, which can be used by networking extensions to easily get the networking actions done.

Template Parameters

<i>T1</i>	the targeted networking connection
<i>T2</i>	the type of the data

Type Constraints

***T2* :** [*UNNetworkData*](#)<*T1*>

5.86.2 Constructor & Destructor Documentation

5.86.2.1 `uNature.Core.Networking.UNNetworkManager< T1, T2 >.UNNetworkManager (MonoBehaviour managerInstance)`

The constructor of this class, initiate basic events.

5.86.3 Member Function Documentation

5.86.3.1 `virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.Awake ()` [virtual]

Called when the network manager is initialized.

5.86.3.2 `IEnumerator uNature.Core.Networking.UNNetworkManager< T1, T2 >.CheckForStreamingBufferedUpdates ()` [protected]

This method is checking every certain amount of seconds for new loaded streamed areas to update data that is waiting for a streamed terrain to be loaded.

Returns

5.86.3.3 `void uNature.Core.Networking.UNNetworkManager< T1, T2 >.OnClientConnected (T1 conn)`

Called when the client connects, send all data.

Parameters

<i>conn</i>	the connection
-------------	----------------

5.86.3.4 `void uNature.Core.Networking.UNNetworkManager< T1, T2 >.OnHarvestableTreeInstancePooled (HarvestableTIPoolItem instance)` [protected]

Called when an harvestable item instance has been created

Parameters

<i>instance</i>	the created instance
-----------------	----------------------

5.86.3.5 void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.OnItemDamaged (HarvestableTIPoolItem *item*, int *damage*) [protected]

Update item damage and handle synchorization.

Parameters

<i>item</i>	
<i>health</i>	

5.86.3.6 virtual void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.SendEvent (UNNetworkData< T1 > *instance*) [virtual]

Send event to the correct location

Parameters

<i>instance</i>	the data instance
-----------------	-------------------

5.86.3.7 virtual void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.SendToClients (UNNetworkData< T1 > *instance*) [virtual]

Send to all of the clients.

Parameters

<i>terrainName</i>	the terrain name (terrain.name)
<i>instanceID</i>	the tree instance
<i>destroy</i>	you want to destroy/ restore the tree?

5.86.3.8 virtual void **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.SendToConnection (T1 *connection*, UNNetworkData< T1 > *instance*) [virtual]

Send to certain connection the data

Parameters

<i>connection</i>	the connection
<i>terrainName</i>	the terrain name (terrain.name)
<i>instanceID</i>	the tree instance
<i>destroy</i>	you want to destroy/ restore the tree?

5.86.3.9 `virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.SendToOthers (UNNetworkData< T1 > instance) [virtual]`

Send to all other connections.

Parameters

<i>terrainName</i>	the terrain name (terrain.name)
<i>instanceID</i>	the tree instance
<i>destroy</i>	you want to destroy/ restore the tree?

5.86.3.10 `virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.SendToServer (UNNetworkData< T1 > instance) [virtual]`

Send to the server the data.

Parameters

<i>terrainName</i>	the terrain name (terrain.name)
<i>instanceID</i>	the tree instance
<i>destroy</i>	you want to destroy/ restore the tree?

5.86.3.11 `virtual void uNature.Core.Networking.UNNetworkManager< T1, T2 >.UpdatePermissions () [virtual]`

Reupdate the permissions for the Pool items. (needs to be called when ever there's a networking change for the owner/controller)

5.86.4 Member Data Documentation

5.86.4.1 `UNNetworkManager<T1, T2> uNature.Core.Networking.UNNetworkManager< T1, T2 >.manager [static]`

a static instance of this object.

5.86.5 Property Documentation

5.86.5.1 `UNList<BaseUNNetworkData> uNature.Core.Networking.UNNetworkManager< T1, T2 >.bufferedData [static], [get]`

The buffered data which will be sent to all of the connecting connections.

5.86.5.2 `virtual bool uNature.Core.Networking.UNNetworkManager< T1, T2 >.isAuth [get]`

is the server architecture is authoritative?

5.86.5.3 virtual bool **uNature.Core.Networking.UNNetworkManager**< T1, T2 >.isServer [get]

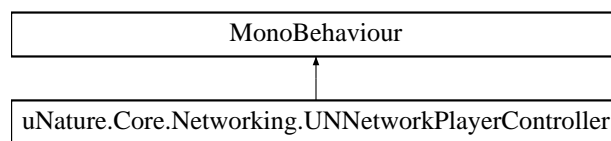
Are we the server?

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/UNNetworkManager.cs

5.87 uNature.Core.Networking.UNNetworkPlayerController Class Reference

Inheritance diagram for uNature.Core.Networking.UNNetworkPlayerController:



Public Member Functions

- virtual void **OnAttached** ()
- void **ManageEnableOnProxies** (bool value)

Public Attributes

- MonoBehaviour[] **disableOnProxies**
- Camera **Camera**
- CharacterController **controller**

Protected Member Functions

- virtual void **Awake** ()

Properties

- virtual bool **hasControl** [get]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Networking_Templates/UNNetworkPlayerController.cs

5.88 uNature.Core.UNPhysicsHit_Grass Struct Reference

A class that holds the data for the hit data

Public Attributes

- Vector3 **point**
- float **distance**

5.88.1 Detailed Description

A class that holds the data for the hit data

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysics.cs

5.89 uNature.Core.UNPhysicsHitsArray Class Reference

An custom array that holds all ray results in an array

Public Member Functions

- void **AddToList** ([UNPhysicsHit_Grass](#) hit)
- void **Sort** ()

Properties

- [UNPhysicsHit_Grass](#) **this[int index]** [get]
- int **Count** [get]

5.89.1 Detailed Description

An custom array that holds all ray results in an array

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysics.cs

5.90 uNature.Core.UNPhysicsObject Struct Reference

This is a base class for a UCPysicsObject. Every class that inherits this class will be counted in the physics system.

Public Member Functions

- void [UpdateBounds](#) ()
Update object's bounds
- void [OnDrawGizmos](#) ()
Draw gizmos
- void [DrawShape](#) (Matrix4x4 matrix)
Draw the shape of the bounds
- bool [Raycast](#) (Ray ray, out [UNPhysicsHit_Grass](#) _hit, LayerMask mask)
Raycast the physics object

Static Public Attributes

- static [UNPhysicsHit_Grass](#) **hit** = new [UNPhysicsHit_Grass](#)()

Properties

- bool **enabled** [get, set]

5.90.1 Detailed Description

This is a base class for a UCPysicsObject. Every class that inherits this class will be counted in the physics system.

5.90.2 Member Function Documentation

5.90.2.1 void uNature.Core.UNPhysicsObject.DrawShape (Matrix4x4 *matrix*)

Draw the shape of the bounds

Parameters

<i>matrix</i>	the matrix of the bounds
<i>selected</i>	is the shape selected in heirachy

5.90.2.2 void uNature.Core.UNPhysicsObject.OnDrawGizmos ()

Draw gizmos

5.90.2.3 bool uNature.Core.UNPhysicsObject.Raycast (Ray *ray*, out [UNPhysicsHit_Grass](#) _hit, LayerMask *mask*)

Raycast the physics object

Parameters

<i>origin</i>	ray origin
<i>direction</i>	ray direction
<i>_hit</i>	hit data
<i>distance</i>	max distance
<i>mask</i>	layerMask

Returns

Did we hit something?

5.90.2.4 void uNature.Core.UNPhysicsObject.UpdateBounds ()

Update object's bounds

Parameters

<i>center</i>	The center of the bounds, worldspace
<i>size</i>	The size of the bounds, worldspace

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysicsObject.cs

5.91 uNature.Core.UNPhysicsTemplate Struct Reference

Public Member Functions

- **UNPhysicsTemplate** (Vector3 position, float spreadX, float spreadZ, int densityIndex, [FoliagePrototype](#) prototype)

Public Attributes

- Vector3 **position**
- Vector2 **spread**
- int **densityIndex**
- [FoliagePrototype](#) **prototype**

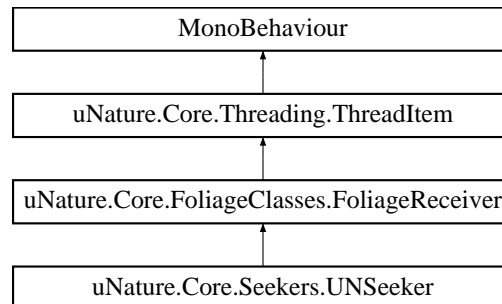
The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Physics/UNPhysicsTemplate.cs

5.92 uNature.Core.Seekers.UNSeeker Class Reference

[Seekers](#) are basically GameObjects in the scene which should interact with the objects in the game.

Inheritance diagram for uNature.Core.Seekers.UNSeeker:



Public Member Functions

- virtual IEnumerator [Start](#) ()
Called on start, initiate initial check targets.

Public Attributes

- float [seekingDistance](#) = 20f
How far will it look ?
- bool [attackTrees](#) = true
Disable this if you want to do your own trees logic.
- int [raycastMask](#) = 1
Ignore layer for tree attack.
- float [raycastDistance](#) = 10
Raycast range for tree attack.

Protected Member Functions

- override void [Update](#) ()
Check for movement.
- virtual void [CheckTargetsOnMove](#) ()
Checks the targets when the character moved enoughed.
- virtual void [HarvestChecks](#) ()
Try to harvest trees.

Protected Attributes

- Vector3 [lastMovement](#) = Vector3.zero
What was the last position our AOI was updated on?
- float [_treesCheckDistance](#) = 10f
After how much distance will it update the trees.

Properties

- float **treesCheckDistance** [get, set]

Additional Inherited Members

5.92.1 Detailed Description

[Seekers](#) are basically GameObjects in the scene which should interact with the objects in the game.

5.92.2 Member Function Documentation

5.92.2.1 virtual void uNature.Core.Seekers.UNSeeker.CheckTargetsOnMove () [protected],[virtual]

Checks the targets when the character moved enough.

5.92.2.2 virtual void uNature.Core.Seekers.UNSeeker.HarvestChecks () [protected],[virtual]

Try to harvest trees.

5.92.2.3 virtual IEnumerator uNature.Core.Seekers.UNSeeker.Start () [virtual]

Called on start, initiate initial check targets.

5.92.2.4 override void uNature.Core.Seekers.UNSeeker.Update () [protected],[virtual]

Check for movement.

Reimplemented from [uNature.Core.FoliageClasses.FoliageReceiver](#).

5.92.3 Member Data Documentation

5.92.3.1 float uNature.Core.Seekers.UNSeeker._treesCheckDistance = 10f [protected]

After how much distance will it update the trees.

5.92.3.2 bool uNature.Core.Seekers.UNSeeker.attackTrees = true

Disable this if you want to do your own trees logic.

5.92.3.3 `Vector3 uNature.Core.Seekers.UNSeeker.lastMovement = Vector3.zero` [protected]

What was the last position our AOI was updated on?

5.92.3.4 `float uNature.Core.Seekers.UNSeeker.raycastDistance = 10`

Raycast range for tree attack.

5.92.3.5 `int uNature.Core.Seekers.UNSeeker.raycastMask = 1`

Ignore layer for tree attack.

5.92.3.6 `float uNature.Core.Seekers.UNSeeker.seekingDistance = 20f`

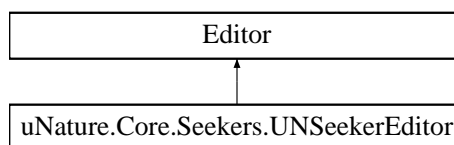
How far will it look ?

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Seekers/UNSeeker.cs

5.93 uNature.Core.Seekers.UNSeekerEditor Class Reference

Inheritance diagram for uNature.Core.Seekers.UNSeekerEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNSeekerEditor.cs

5.94 uNature.Core.Settings.UNSetting Class Reference

A class which should be used on custom classes that needs to be shown and serialized as a setting.

Public Member Functions

- virtual void [DrawGUI](#) ()

Draw the gui of the setting on this method. This will be called from the [UNSettingsEditor](#).

5.94.1 Detailed Description

A class which should be used on custom classes that needs to be shown and serialized as a setting.

5.94.2 Member Function Documentation

5.94.2.1 virtual void uNature.Core.Settings.UNSetting.DrawGUI () [virtual]

Draw the gui of the setting on this method. This will be called from the [UNSettingsEditor](#).

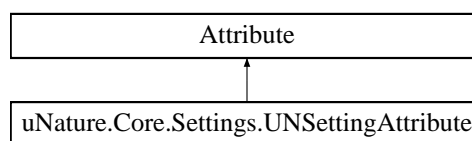
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSetting.cs

5.95 uNature.Core.Settings.UNSettingAttribute Class Reference

The attribute of each setting which handles the drawing of the setting (generically).

Inheritance diagram for uNature.Core.Settings.UNSettingAttribute:



Public Member Functions

- **UNSettingAttribute** ([UNSettingCategories](#) category, string name)
- **UNSettingAttribute** ([UNSettingCategories](#) category, string name, string desc)
- object **Draw** (object instance)

Public Attributes

- [UNSettingCategories](#) **category**
- string **name**
- string **desc**

5.95.1 Detailed Description

The attribute of each setting which handles the drawing of the setting (generically).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSettings.cs

5.96 uNature.Core.Settings.UNSettingCategory Class Reference

The class of a category which handles keeping hold of all of the categories and makes all of the reflection needed.

Public Member Functions

- **UNSettingCategory** ([UNSettingCategories](#) category)

Static Public Member Functions

- static [UNSettingCategory](#) **GetCategory** ([UNSettingCategories](#) category)

Public Attributes

- bool **show**
- [UNSettingCategories](#) **type**
- List< [UNSettingAttribute](#) > **attributes** = new List<[UNSettingAttribute](#)>()
- List< FieldInfo > **fields** = new List<FieldInfo>()

Properties

- static List< [UNSettingCategory](#) > **categories** [get]

5.96.1 Detailed Description

The class of a category which handles keeping hold of all of the categories and makes all of the reflection needed.

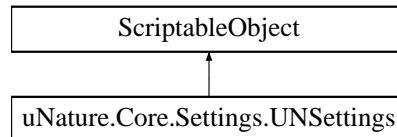
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSettings.cs

5.97 uNature.Core.Settings.UNSettings Class Reference

A class which handles certain settings of aspects in UN

Inheritance diagram for uNature.Core.Settings.UNSettings:



Public Member Functions

- void [ResetDefaults](#) ()
Reset the settings to the default state.

Static Public Member Functions

- static void [Log](#) (string context)
Log a message on the [uNature](#) debug mode.

Public Attributes

- const string [ProjectVersion](#) = "2.1"
The version number of this package.
- const string [fileName](#) = "UNSettings"
The file name which will be created for this settings file.
- const string [ProjectName](#) = "uNature"
Project name (UN folder name).
- bool [UN_TreeInstancesRespawnsEnabled](#) = false
- bool [UN_Networking_Auth](#) = true
- float [UN_TreeInstancesRespawnsTime](#) = 1
- bool [UN_Threading_Enabled](#) = true
- Threading.uNature_Thread_Workers [UN_Threading_WorkersCount](#) = Threading.uNature_Thread_↔
Workers.One_Worker
- bool [UN_Debugging_Enabled](#) = false
- bool [UN_Console_Debugging_Enabled](#) = false
- bool [UN_Foliage_RUNTIME_SAVECHANGES](#) = false

Properties

- static string [ProjectPath](#) [get]
The found path to the project directory (based on the name provided on ProjectName).
- static [UNSettings instance](#) [get]

5.97.1 Detailed Description

A class which handles certain settings of aspects in UN

5.97.2 Member Function Documentation

5.97.2.1 static void uNature.Core.Settings.UNSettings.Log (string context) [static]

Log a message on the [uNature](#) debug mode.

Parameters

<i>context</i>	
----------------	--

5.97.2.2 void uNature.Core.Settings.UNSettings.ResetDefaults ()

Reset the settings to the default state.

5.97.3 Member Data Documentation

5.97.3.1 const string uNature.Core.Settings.UNSettings.fileName = "UNSettings"

The file name which will be created for this settings file.

5.97.3.2 const string uNature.Core.Settings.UNSettings.ProjectName = "uNature"

Project name (UN folder name).

5.97.3.3 const string uNature.Core.Settings.UNSettings.ProjectVersion = "2.1"

The version number of this package.

5.97.4 Property Documentation

5.97.4.1 string uNature.Core.Settings.UNSettings.ProjectPath [static], [get]

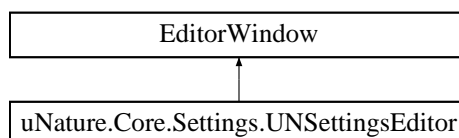
The found path to the project directory (based on the name provided on ProjectName).

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/UNSettings.cs

5.98 uNature.Core.Settings.UNSettingsEditor Class Reference

Inheritance diagram for uNature.Core.Settings.UNSettingsEditor:



Static Public Member Functions

- static void **Open** ()

Public Attributes

- [UNSettings](#) **_settings**

Properties

- [UNSettings](#) **settings** [get]

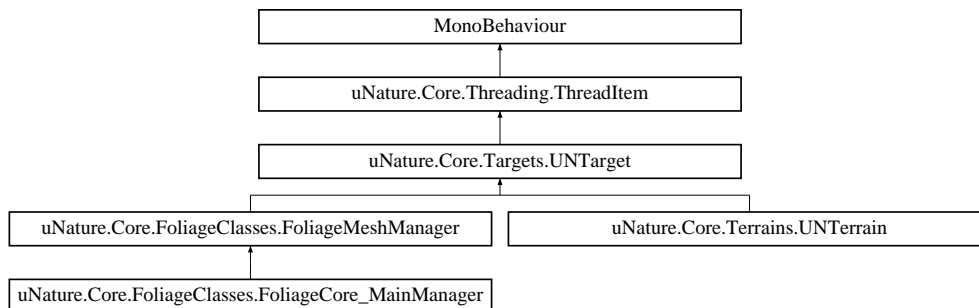
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Settings/Editor/UNSettingsEditor.cs

5.99 uNature.Core.Targets.UNTarget Class Reference

A target is what will be taken into account with the system. For example terrains.

Inheritance diagram for uNature.Core.Targets.UNTarget:



Public Member Functions

- virtual void **Check** ([Seekers.UNSeeker](#) seeker, Vector3 seekerPos, float seekingDistance, bool isPlaying)
Check and apply AOI from seeker.

Parameters

seeker	<i>Our seeker</i>
seekerPos	<i>the seeker position -> in order to maintain multithreading.</i>

- virtual bool **InDistance** ([UNSeeker](#) seeker)
Confirm that a seeker is in the range of the target.
- virtual Vector3 **FixPosition** (Vector3 position)
Fix the position that is given to the local space position of this target - for example in the terrain you want to reduce the terrain position.

- virtual void [OnDrawGizmos](#) ()
Draw gizmos
- virtual void [CreatePool](#) (System.Type PoolItemType)
Create Pool.

Static Public Member Functions

- static void [CheckTargets](#) ([UNSeeker](#) seeker, float distance)
Check and apply aoi from a certain seeekr.

Public Attributes

- [Pool](#) [Pool](#)
A Pool which is used to increase performance on runtime, which manages objects smartly than instantiating them manually on runtime each time.
- string [PoolTypeSerializedName](#) = ""
Was the Pool type de-serialized.
- int [PoolAmount](#) = 15
How many objects will be created for each Pool type.

Static Public Attributes

- static List< [UNTarget](#) > [worldTargets](#) = new List<[UNTarget](#)>()
All of the targets in the scene.

Protected Member Functions

- virtual void [Awake](#) ()
Initiate awake settings.
- override void [Update](#) ()
Called every frame
- override void [OnEnable](#) ()
Add this target to the targets Pool
- override void [OnDisable](#) ()
Remove this target to the targets Pool

Properties

- System.Type **PoolItemType** [get, set]
- virtual bool [useMultithreadedCheck](#) [get]
Will the system call a multi-threaded task for making the checks ?

5.99.1 Detailed Description

A target is what will be taken into account with the system. For example terrains.

5.99.2 Member Function Documentation

5.99.2.1 virtual void uNature.Core.Targets.UNTarget.Awake () [protected],[virtual]

Initiate awake settings.

Reimplemented in [uNature.Core.Terrains.UNTerrain](#), and [uNature.Core.FoliageClasses.FoliageCore_MainManager](#).

5.99.2.2 virtual void uNature.Core.Targets.UNTarget.Check (Seekers.UNSeeker *seeker*, Vector3 *seekerPos*, float *seekingDistance*, bool *isPlaying*) [virtual]

Check and apply AOI from seeker.

Parameters

<i>seeker</i>	Our seeker
<i>seekerPos</i>	the seeker position -> in order to maintain multithreading.

5.99.2.3 static void uNature.Core.Targets.UNTarget.CheckTargets (UNSeeker *seeker*, float *distance*) [static]

Check and apply aoi from a certain seekr.

Parameters

<i>seeker</i>	our seeker.
<i>distance</i>	seeking distance

5.99.2.4 virtual void uNature.Core.Targets.UNTarget.CreatePool (System.Type *PoolItemType*) [virtual]

Create Pool.

Reimplemented in [uNature.Core.Terrains.UNTerrain](#).

5.99.2.5 virtual Vector3 uNature.Core.Targets.UNTarget.FixPosition (Vector3 *position*) [virtual]

Fix the position that is given to the local space position of this target - for example in the terrain you want to reduce the terrain position.

Parameters

<i>position</i>	the position
-----------------	--------------

Returns

fixed position

Reimplemented in [uNature.Core.Terrains.UNTerrain](#).

5.99.2.6 virtual bool uNature.Core.Targets.UNTarget.InDistance (UNSeeker *seeker*) [virtual]

Confirm that a seeker is in the range of the target.

Parameters

<i>seeker</i>	The seeker.
---------------	-------------

Returns

Is the inrange of our target?

5.99.2.7 override void uNature.Core.Targets.UNTarget.OnDisable () [protected],[virtual]

Remove this target to the targets Pool

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

Reimplemented in [uNature.Core.Terrains.UNTerrain](#).

5.99.2.8 virtual void uNature.Core.Targets.UNTarget.OnDrawGizmos () [virtual]

Draw gizmos

5.99.2.9 override void uNature.Core.Targets.UNTarget.OnEnable () [protected],[virtual]

Add this target to the targets Pool

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

Reimplemented in [uNature.Core.Terrains.UNTerrain](#).

5.99.2.10 override void uNature.Core.Targets.UNTarget.Update () [protected],[virtual]

Called every frame

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.99.3 Member Data Documentation

5.99.3.1 Pool uNature.Core.Targets.UNTarget.Pool

A Pool which is used to increase performance on runtime, which manages objects smartly than instantiating them manually on runtime each time.

5.99.3.2 int uNature.Core.Targets.UNTarget.PoolAmount = 15

How many objects will be created for each Pool type.

5.99.3.3 string uNature.Core.Targets.UNTarget.PoolTypeSerializedName = ""

Was the Pool type de-serialized.

5.99.3.4 List<UNTarget> uNature.Core.Targets.UNTarget.worldTargets = new List<UNTarget>() [static]

All of the targets in the scene.

5.99.4 Property Documentation

5.99.4.1 virtual bool uNature.Core.Targets.UNTarget.useMultithreadedCheck [get], [protected]

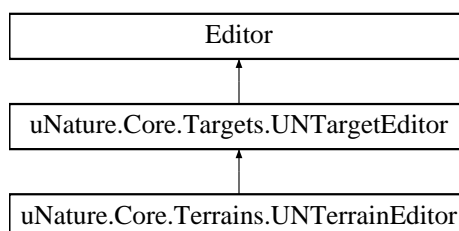
Will the system call a multi-threaded task for making the checks ?

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Targets/UNTarget.cs

5.100 uNature.Core.Targets.UNTargetEditor Class Reference

Inheritance diagram for uNature.Core.Targets.UNTargetEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

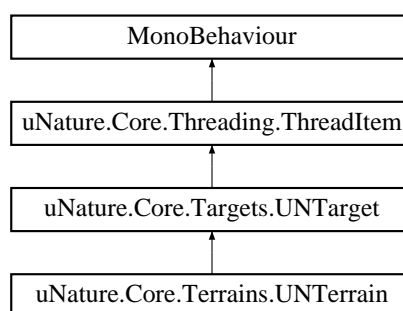
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNTargetEditor.cs

5.101 uNature.Core.Terrains.UNTerrain Class Reference

A class that needs to be on each terrain that needs to be taken into account when managing the system.

Inheritance diagram for uNature.Core.Terrains.UNTerrain:



Public Member Functions

- virtual **UNTerrainSector GenerateSector** (int **sectorResolution**, bool multiThread)
*Generate a sector and assign it to the **UNTerrain**.*
- virtual **UNTerrainSector GenerateSector** (int **sectorResolution**)
*Generate a sector and assign it to the **UNTerrain**.*
- override void **Check** (**Seekers.UNSeeker** seeker, Vector3 seekerPos, float seekingDistance, bool isPlaying)
Check for seeker on terrain.
- override bool **InDistance** (**Seekers.UNSeeker** seeker)
Check if the seeker is in range of the terrain.
- override void **CreatePool** (System.Type PoolItemType)
Fill up our Pool.
- override Vector3 **FixPosition** (Vector3 position)
Return the position that can be used with the chunks.

Public Attributes

- const float **removedTreeInstanceHeight** = 0f
The height which "destroyed" tree instances will get. Don't change if not needed.
- const int **collidersPoolItemInstanceIncrease** = 10000
The height which "destroyed" tree instances will get. Don't change if not needed.
- float **distanceOffset**
By how much will the terrain distance be still considered?
- Vector3 **lastSceneViewPosition** = Vector3.zero
- IEnumerator **verifyTreeInstancesChangeRoutine**
An routine that is used from the editor to perform realtime tree instances updates.

Static Public Attributes

- static List< [UNTerrain](#) > [terrains](#) = new List<[UNTerrain](#)>()
All of the current existing terrains in the current scene.

Protected Member Functions

- override void [Awake](#) ()
Initiate startup variables
- virtual void [OnTerrainChanged](#) (int changedFlags)
On terrain changed.
- virtual void [CheckForTreeInstancesRespawns](#) ()
This method will check every set amount of time the trees in the terrain and restore them if needed.
- override void [OnEnable](#) ()
Add this terrain to the terrains Pool
- override void [OnDisable](#) ()
Remove this terrain to the terrains Pool
- override void [OnPositionChanged](#) (Vector3 newPosition)
On terrain position changed

Properties

- Terrain **terrain** [get, set]
- [UNTerrainData](#) **terrainData** [get]
- [UNTerrainSector](#) **sector** [get, set]
- int [sectorResolution](#) [get, set]
How much times will the terrain be divided? the more => the slower creation but higher performance on runtime.. the less => faster creation but lower performance on runtime.
- bool [manageGrass](#) [get, set]
Will the system try to optimize your grass?
- bool [updateGrassOnHeightsChange](#) [get, set]
- bool [manageTrees](#) [get, set]
Will the system try to optimize your trees?

Additional Inherited Members

5.101.1 Detailed Description

A class that needs to be on each terrain that needs to be taken into account when managing the system.

5.101.2 Member Function Documentation

5.101.2.1 override void uNature.Core.Terrains.UNTerrain.Awake () [protected],[virtual]

Initiate startup variables

Reimplemented from [uNature.Core.Targets.UNTarget](#).

5.101.2.2 override void uNature.Core.Terrains.UNTerrain.Check (Seekers.UNSeeker seeker, Vector3 seekerPos, float seekingDistance, bool isPlaying)

Check for seeker on terrain.

Parameters

<i>seeker</i>	Our seeker.
<i>seekerPos</i>	the seeker pos

5.101.2.3 `virtual void uNature.Core.Terrains.UNTerrain.CheckForTreeInstancesRespawns () [protected], [virtual]`

This method will check every set amount of time the trees in the terrain and restore them if needed.

5.101.2.4 `override void uNature.Core.Terrains.UNTerrain.CreatePool (System.Type PoolItemType) [virtual]`

Fill up our Pool.

Reimplemented from [uNature.Core.Targets.UNTarget](#).

5.101.2.5 `override Vector3 uNature.Core.Terrains.UNTerrain.FixPosition (Vector3 position) [virtual]`

Return the position that can be used with the chunks.

Parameters

<i>position</i>	the original position
-----------------	-----------------------

Returns

position that can be used in local space with the terrain

Reimplemented from [uNature.Core.Targets.UNTarget](#).

5.101.2.6 `virtual UNTerrainSector uNature.Core.Terrains.UNTerrain.GenerateSector (int sectorResolution, bool multiThread) [virtual]`

Generate a sector and assign it to the [UNTerrain](#).

Parameters

<i>sectorResolution</i>	How many pieces will the terrain be divided to? the bigger it is the more pieces.
-------------------------	---

5.101.2.7 `virtual UNTerrainSector uNature.Core.Terrains.UNTerrain.GenerateSector (int sectorResolution) [virtual]`

Generate a sector and assign it to the [UNTerrain](#).

Parameters

<i>sectorResolution</i>	How many pieces will the terrain be divided to? the bigger it is the more pieces.
-------------------------	---

5.101.2.8 override bool uNature.Core.Terrains.UNTerrain.InDistance (**Seekers.UNSeeker** *seeker*)

Check if the seeker is in range of the terrain.

Parameters

<i>seeker</i>	Seeker
---------------	--------

Returns

in range?

5.101.2.9 override void uNature.Core.Terrains.UNTerrain.OnDisable () [protected],[virtual]

Remove this terrain to the terrains Pool

Reimplemented from [uNature.Core.Targets.UNTarget](#).

5.101.2.10 override void uNature.Core.Terrains.UNTerrain.OnEnable () [protected],[virtual]

Add this terrain to the terrains Pool

Reimplemented from [uNature.Core.Targets.UNTarget](#).

5.101.2.11 override void uNature.Core.Terrains.UNTerrain.OnPositionChanged (**Vector3** *newPosition*) [protected],[virtual]

On terrain position changed

Parameters

<i>newPosition</i>	
--------------------	--

Reimplemented from [uNature.Core.Threading.ThreadItem](#).

5.101.2.12 virtual void uNature.Core.Terrains.UNTerrain.OnTerrainChanged (int *changedFlags*) [protected],[virtual]

On terrain changed.

Parameters

<i>changedFlags</i>	
---------------------	--

5.101.3 Member Data Documentation

5.101.3.1 `const int uNature.Core.Terrains.UNTerrain.collidersPoolItemInstanceIncrease = 10000`

The height which "destroyed" tree instances will get. Don't change if not needed.

5.101.3.2 `float uNature.Core.Terrains.UNTerrain.distanceOffset`

By how much will the terrain distance be still considered?

5.101.3.3 `const float uNature.Core.Terrains.UNTerrain.removedTreeInstanceHeight = 0f`

The height which "destroyed" tree instances will get. Don't change if not needed.

5.101.3.4 `List<UNTerrain> uNature.Core.Terrains.UNTerrain.terrains = new List<UNTerrain>() [static]`

All of the current existing terrains in the current scene.

5.101.3.5 `IEnumerator uNature.Core.Terrains.UNTerrain.verifyTreeInstancesChangeRoutine`

An routine that is used from the editor to perform realtime tree instances updates.

5.101.4 Property Documentation

5.101.4.1 `bool uNature.Core.Terrains.UNTerrain.manageGrass [get], [set]`

Will the system try to optimize your grass?

Also, make sure to design the grass LODs if the grass doesnt work as you'd like (Window/uNature/Settings).

5.101.4.2 `bool uNature.Core.Terrains.UNTerrain.manageTrees [get], [set]`

Will the system try to optimize your trees?

5.101.4.3 int uNature.Core.Terrains.UNTerrain.sectorResolution [get], [set]

How much times will the terrain be divided? the more => the slower creation but higher performance on runtime.. the less => faster creation but lower performance on runtime.

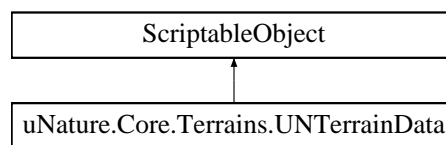
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/UNTerrain.cs

5.102 uNature.Core.Terrains.UNTerrainData Class Reference

The terrain data class which is used by [uNature](#). Can be accessed by "UNTerrain.terrainData".

Inheritance diagram for uNature.Core.Terrains.UNTerrainData:



Public Member Functions

- [UNTreePrototype GetPrototype](#) (TreePrototype prototype)
Get a terrain tree prototype ([UNTerrainData](#)) from a unity's tree prototype.
- void [UpdateMultithreadedVariables](#) ()
Update the current multi-threaded variables on this terrain so it can be used on a different thread.
- void [Backup](#) ()
Backup the terrain data.
- void [ApplyBackup](#) (Terrain terrain)
Apply the current backup.
- void [DeleteBackup](#) ()
Delete the current backup.
- void [Initialize](#) ()
Initialize the terrain data.

Static Public Member Functions

- static string [GetBackUpPath](#) (string backUpName)
- static [UNTerrainData GetInstance](#) (TerrainData terrainData)
Get the [UNTerrainData](#) instance from providing a terrainData.

Public Attributes

- Vector2 [FoliageRelativeMultiplier](#)
- int [multiThreaded_detailResolution](#)
- int [multiThreaded_detailWidth](#)
- int [multiThreaded_detailHeight](#)
- Vector3[,] [multiThreaded_terrainSampleNormals](#)
- DetailPrototype[] [multiThreaded_detailPrototypes](#)
- float [multiThreaded_heightMapWidth](#)
- float [multiThreaded_heightMapHeight](#)

Protected Member Functions

- virtual void [CheckForTreePrototypesChange](#) ()
Checks and updates dirty terrain tree prototypes.
- void [SendUpdateEventToLinkedTerrains](#) (TerrainChangedFlags flag)
Send an TerrainData changed event to all linked terrains.

Properties

- static string **persistentPath** [get]
- string **objectName** [get, set]
- TerrainData [terrainData](#) [get, protected set]
The terrain data which this object resembles
- TerrainData **backedUpTerrainData** [get]
- List< [UNTreePrototype](#) > **treePrototypes** [get]
- float[,] [heights](#) [get]
Get the heights of the terrain (used in multi-threading).
- bool **isDirty** [get]
- Vector3 **multiThreaded_terrainDataSize** [get, set]
- float[,] **multiThreaded_terrainHeights** [get, set]

5.102.1 Detailed Description

The terrain data class which is used by [uNature](#). Can be accessed by "UNTerrain.terrainData".

5.102.2 Member Function Documentation

5.102.2.1 void uNature.Core.Terrains.UNTerrainData.ApplyBackup (Terrain *terrain*)

Apply the current backup.

5.102.2.2 void uNature.Core.Terrains.UNTerrainData.Backup ()

Backup the terrain data.

5.102.2.3 virtual void uNature.Core.Terrains.UNTerrainData.CheckForTreePrototypesChange () [protected], [virtual]

Checks and updates dirty terrain tree prototypes.

5.102.2.4 void uNature.Core.Terrains.UNTerrainData.DeleteBackup ()

Delete the current backup.

5.102.2.5 static UNTerrainData uNature.Core.Terrains.UNTerrainData.GetInstance (TerrainData *terrainData*) [static]

Get the [UNTerrainData](#) instance from providing a terrainData.

Parameters

<i>terrainData</i>	the terrain data which this UNTerrainData belongs to.
--------------------	---

Returns

the [UNTerrainData](#) instance for the provided terrain data.

5.102.2.6 `UNTreePrototype uNature.Core.Terrains.UNTerrainData.GetPrototype (TreePrototype prototype)`

Get a terrain tree prototype ([UNTerrainData](#)) from a unity's tree prototype.

Parameters

<i>prototype</i>	
------------------	--

Returns

5.102.2.7 `void uNature.Core.Terrains.UNTerrainData.Initialize ()`

Initialize the terrain data.

5.102.2.8 `void uNature.Core.Terrains.UNTerrainData.SendUpdateEventToLinkedTerrains (TerrainChangedFlags flag)` [protected]

Send an TerrainData changed event to all linked terrains.

5.102.2.9 `void uNature.Core.Terrains.UNTerrainData.UpdateMultithreadedVariables ()`

Update the current multi-threaded variables on this terrain so it can be used on a different thread.

5.102.3 Property Documentation

5.102.3.1 `float [,] uNature.Core.Terrains.UNTerrainData.heights` [get]

Get the heights of the terrain (used in multi-threading).

5.102.3.2 `TerrainData uNature.Core.Terrains.UNTerrainData.terrainData` [get], [protected set]

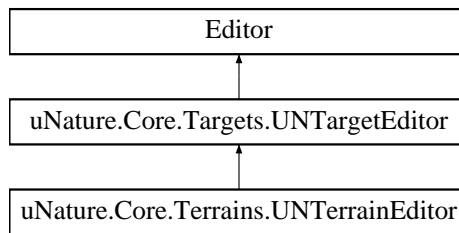
The terrain data which this object resembles

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/TerrainData/UNTerrainData.cs

5.103 uNature.Core.Terrains.UNTerrainEditor Class Reference

Inheritance diagram for uNature.Core.Terrains.UNTerrainEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

Public Attributes

- List< [UNTreePrototype](#) > **selectedPrototypes** = new List<[UNTreePrototype](#)>()
- TerrainTabs **currentTab** = TerrainTabs.Grids

Properties

- List< SelectionBoxItems< System.Type > > **PoolCache** [get]

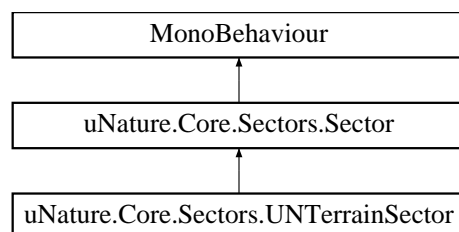
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Editor/UNTerrainEditor.cs

5.104 uNature.Core.Sectors.UNTerrainSector Class Reference

An sector class dedicated only for terrains.

Inheritance diagram for uNature.Core.Sectors.UNTerrainSector:



Public Member Functions

- override void [OnCreated](#) (Transform owner, int resolution)
Called when the object is created.

Parameters

terrain	The terrain we belong to.
---------	---------------------------

- override void [Awake](#) ()
Called on awake.
- void [FetchTreeInstances](#) (bool useUNThread, System.Action OnFinish)
Get all the terrain tree instances into chunks

Parameters

useUNThread	Do you want to use the uNature thread to reduce performance issues ?
-------------	--

- override void [ApplicationQuit](#) ()
Called when the application has quit.

Public Attributes

- [UNTerrain](#) **unTerrain**
- Terrain **_terrain**
- TreeInstance[] **originalTreeInstances**

Protected Member Functions

- override void [OnChunkCreated](#) ([Chunk](#) chunk)
Called when a chunk is created to allow custom logic on the inherited sectors.
- override void [OnStartCreatingChunks](#) ()
Called right before starting to create the chunks.

Properties

- Terrain **terrain** [get, set]
- int **treeInstancesCount** [get, set]
- List< [TChunk](#) > **treeInstancesChunks** [get]
- bool **restoreComplete** [get]

Additional Inherited Members**5.104.1 Detailed Description**

An sector class dedicated only for terrains.

5.104.2 Member Function Documentation**5.104.2.1** override void uNature.Core.Sectors.UNTerrainSector.ApplicationQuit () [virtual]

Called when the application has quit.

Reimplemented from [uNature.Core.Sectors.Sector](#).

5.104.2.2 `override void uNature.Core.Sectors.UNTerrainSector.Awake () [virtual]`

Called on awake.

Reimplemented from [uNature.Core.Sectors.Sector](#).

5.104.2.3 `void uNature.Core.Sectors.UNTerrainSector.FetchTreeInstances (bool useUNThread, System.Action OnFinish)`

Get all the terrain tree instances into chunks

Parameters

<i>useUNThread</i>	Do you want to use the uNature thread to reduce performance issues ?
--------------------	--

5.104.2.4 `override void uNature.Core.Sectors.UNTerrainSector.OnChunkCreated (Chunk chunk) [protected], [virtual]`

Called when a chunk is created to allow custom logic on the inherited sectors.

Parameters

<i>chunk</i>	
--------------	--

Reimplemented from [uNature.Core.Sectors.Sector](#).

5.104.2.5 `override void uNature.Core.Sectors.UNTerrainSector.OnCreated (Transform owner, int resolution) [virtual]`

Called when the object is created.

Parameters

<i>terrain</i>	The terrain we belong to.
----------------	---------------------------

Reimplemented from [uNature.Core.Sectors.Sector](#).

5.104.2.6 `override void uNature.Core.Sectors.UNTerrainSector.OnStartCreatingChunks () [protected], [virtual]`

Called right before starting to create the chunks.

Reimplemented from [uNature.Core.Sectors.Sector](#).

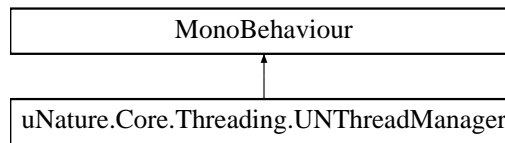
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/UNTerrainSector.cs

5.105 uNature.Core.Threading.UNThreadManager Class Reference

This class handles the multi-threading mechanics.

Inheritance diagram for uNature.Core.Threading.UNThreadManager:



Public Member Functions

- void `UpdateThreadItems` ()
Updates the thread items in the scene.
- void `RunOnUnityThread` (`IThreadTask` action)
Add an action to the unity thread
- void `RunOnThread` (`IThreadTask` action)
Add an action to the UN thread
- void `DelayActionSeconds` (`IThreadTask` task, float time)
Run any action with a specific delay of seconds.
- void `DelayActionFrames` (int frames, `IThreadTask` task)
Run any action after 1 frame

Static Public Member Functions

- static void `InitializelfNotAvailable` ()

Static Public Attributes

- static float `updateThreadItemsTime` = 0.1f
How often will the thread manager update the thread items.

Protected Member Functions

- void `OnThreadProcess` (System.Object processObject)
Called when the thread needs to process the task.

Properties

- static `UNThreadManager instance` [get]
- bool `threadEnabled` [get]
Is the multi-thread option enabled?
- int `threadWorkersCount` [get]
Thread workers count
- static bool `inUnityThread` [get]

5.105.1 Detailed Description

This class handles the multi-threading mechanics.

5.105.2 Member Function Documentation

5.105.2.1 void uNature.Core.Threading.UNThreadManager.DelayActionFrames (int *frames*, IThreadTask *task*)

Run any action after 1 frame

Parameters

<i>task</i>	the task you want to run after 1 frame
-------------	--

5.105.2.2 void uNature.Core.Threading.UNThreadManager.DelayActionSeconds (IThreadTask *task*, float *time*)

Run any action with a specific delay of seconds.

Parameters

<i>task</i>	the task you want to run after the specific amount of seconds
<i>time</i>	the specific amount of seconds to wait

5.105.2.3 void uNature.Core.Threading.UNThreadManager.OnThreadProcess (System.Object *processObject*) [protected]

Called when the thread needs to process the task.

Parameters

<i>processObject</i>	
----------------------	--

5.105.2.4 void uNature.Core.Threading.UNThreadManager.RunOnThread (IThreadTask *action*)

Add an action to the UN thread

Parameters

<i>action</i>	the action
---------------	------------

5.105.2.5 void uNature.Core.Threading.UNThreadManager.RunOnUnityThread (IThreadTask *action*)

Add an action to the unity thread

Parameters

<i>action</i>	the action
---------------	------------

5.105.2.6 void uNature.Core.Threading.UNThreadManager.UpdateThreadItems ()

Updates the thread items in the scene.

5.105.3 Member Data Documentation

5.105.3.1 float uNature.Core.Threading.UNThreadManager.updateThreadItemsTime = 0.1f [static]

How often will the thread manager update the thread items.

5.105.4 Property Documentation

5.105.4.1 bool uNature.Core.Threading.UNThreadManager.threadEnabled [get]

Is the multi-thread option enabled?

5.105.4.2 int uNature.Core.Threading.UNThreadManager.threadWorkersCount [get]

Thread workers count

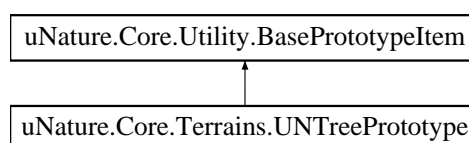
The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/MultiThreading/UNThreadManager.cs

5.106 uNature.Core.Terrains.UNTreePrototype Class Reference

A custom class for the normal tree prototypes. Holds custom data that is used over this certain terrain data.

Inheritance diagram for uNature.Core.Terrains.UNTreePrototype:



Public Member Functions

- [UNTreePrototype](#) (TreePrototype prototype)
Create a new instance of the object
- override bool [Equals](#) (object obj)
Custom equals operator to take into account treePrototypes.
- override int [GetHashCode](#) ()
Override to avoid warnings.

Static Public Member Functions

- static void [CheckForMissings](#) (List< [UNTreePrototype](#) > items, TreePrototype[] prototypes)
This method will check whether any of the items is missing.

Public Attributes

- GameObject [prototypeObject](#)
the game object of the tree prototype.
- bool [isMissing](#) = false
Is this prototype missing on the terrainData? if so, make sure to wait for it to "comeback" and meanwhile store its data.
- int [prototypeIndex](#)

Protected Member Functions

- override Texture2D [GetPreview](#) ()
Get Item Preview

Properties

- override bool [isEnabled](#) [get]
Is this prototype missing?
- bool [enabled](#) [get, set]
- bool [forcePoolCreation](#) [get, set]

5.106.1 Detailed Description

A custom class for the normal tree prototypes. Holds custom data that is used over this certain terrain data.

5.106.2 Constructor & Destructor Documentation

5.106.2.1 uNature.Core.Terrains.UNTreePrototype.UNTreePrototype (TreePrototype prototype)

Create a new instance of the object

Parameters

<i>prototype</i>	The tree prototype this instance is based on.
------------------	---

5.106.3 Member Function Documentation

5.106.3.1 static void uNature.Core.Terrains.UNTreePrototype.CheckForMissings (List< UNTreePrototype > *items*, TreePrototype[] *prototypes*) [static]

This method will check whether any of the items is missing.

Parameters

<i>items</i>	the items list
<i>prototypes</i>	the tree prototypes of the terrain data

5.106.3.2 override bool uNature.Core.Terrains.UNTreePrototype.Equals (object *obj*)

Custom equals operator to take into account treePrototypes.

Parameters

<i>obj</i>	
------------	--

Returns

5.106.3.3 override int uNature.Core.Terrains.UNTreePrototype.GetHashCode ()

Override to avoid warnings.

Returns

5.106.3.4 override Texture2D uNature.Core.Terrains.UNTreePrototype.GetPreview () [protected],[virtual]

Get Item Preview

Returns

Reimplemented from [uNature.Core.Utility.BasePrototypeItem](#).

5.106.4 Member Data Documentation

5.106.4.1 bool uNature.Core.Terrains.UNTreePrototype.isMissing = false

Is this prototype missing on the terrainData? if so, make sure to wait for it to "comeback" and meanwhile store its data.

5.106.4.2 GameObject uNature.Core.Terrains.UNTreePrototype.prototypeObject

the game object of the tree prototype.

5.106.5 Property Documentation

5.106.5.1 override bool uNature.Core.Terrains.UNTreePrototype.isEnabled [get]

Is this prototype missing?

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Terrain/TerrainData/UNTerrainData.cs

5.107 uNature.Core.Vector2i Struct Reference

Public Member Functions

- **Vector2i** (int x, int y)
- override bool **Equals** (object obj)
- override int **GetHashCode** ()
- override string **ToString** ()

Static Public Member Functions

- static **Vector2i operator+** (**Vector2i** a, **Vector2i** b)
- static **Vector2i operator-** (**Vector2i** a, **Vector2i** b)
- static **Vector2i operator*** (**Vector2i** a, **Vector2i** b)
- static **Vector2i operator/** (**Vector2i** a, **Vector2i** b)
- static bool **operator==** (**Vector2i** a, **Vector2i** b)
- static bool **operator!=** (**Vector2i** a, **Vector2i** b)

Public Attributes

- int **x**
- int **y**

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNMath.cs

5.108 uNature.Core.Utility.Vector3_XZ_FAST Struct Reference

Public Member Functions

- **Vector3_XZ_FAST** (Vector3 v3)
- **Vector3_XZ_FAST** (Vector2 v2)
- **Vector3_XZ_FAST** (float x, float z)

Public Attributes

- float **x**
- float **z**

The documentation for this struct was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Utility/UNStandaloneUtility.cs

5.109 uNature.Core.FoliageClasses.WindSettings Class Reference

Properties

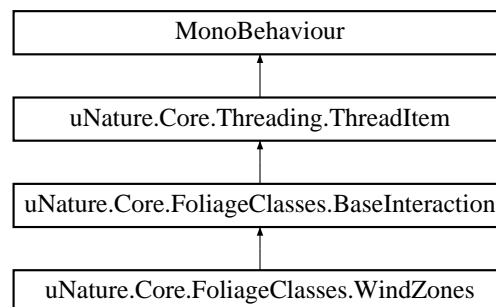
- float **windBending** [get, set]
- float **windSpeed** [get, set]

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/FoliageDB.cs

5.110 uNature.Core.FoliageClasses.WindZones Class Reference

Inheritance diagram for uNature.Core.FoliageClasses.WindZones:



Additional Inherited Members

The documentation for this class was generated from the following file:

- D:/Projects/uNature/Assets/uNature/Scripts/Core/Foilage/Interaction/WindZones.cs

Index

- `_gameObject`
 - `uNature::Core::Pooling::PoolItem`, 81
 - `_terrain`
 - `uNature::Core::Pooling::TerrainPoolItem`, 93
 - `_threadItems`
 - `uNature::Core::Threading::ThreadItem`, 95
 - `_treesCheckDistance`
 - `uNature::Core::Seekers::UNSeeker`, 137
 - `Add`
 - `uNature::Core::Collections::UNList`, 120
 - `AddPrototype`
 - `uNature::Core::FoliageClasses::FoliageDB`, 39
 - `AddToPool`
 - `uNature::Core::Pooling::Pool`, 75
 - `AddTreeInstance`
 - `uNature::Core::Sectors::TICChunk`, 98
 - `ApplicationQuit`
 - `uNature::Core::Sectors::Sector`, 87
 - `uNature::Core::Sectors::UNTerrainSector`, 157
 - `ApplyBackup`
 - `uNature::Core::Terrains::UNTerrainData`, 154
 - `ApplyColorMap`
 - `uNature::Core::FoliageClasses::FoliagePrototype`, 57
 - `ApplyGrassMap`
 - `uNature::Core::FoliageClasses::FoliagePrototype`, 57
 - `ApplyWind`
 - `uNature::Core::FoliageClasses::FoliagePrototype`, 57
 - `AssetDescription`
 - `uNature::Core::Extensions::UNExtension`, 116
 - `AssetDocumentationName`
 - `uNature::Core::Extensions::UNExtension`, 116
 - `AssetLogoName`
 - `uNature::Core::Extensions::UNExtension`, 117
 - `AssetName`
 - `uNature::Core::Extensions::UNExtension`, 117
 - `AssetNameSpace`
 - `uNature::Core::Extensions::UNExtension`, 117
 - `AssetStoreAdress`
 - `uNature::Core::Extensions::UNExtension`, 117
 - `attackTrees`
 - `uNature::Core::Seekers::UNSeeker`, 137
 - `Awake`
 - `uNature::Core::FoliageClasses::FoliageCore_↵`
Chunk, 30
 - `uNature::Core::FoliageClasses::FoliageCore_↵`
MainManager, 33
 - `uNature::Core::Networking::UNNetworkManager`, 129
 - `uNature::Core::Pooling::HarvestableTIPoolItem`, 68
 - `uNature::Core::Pooling::PoolItem`, 80
 - `uNature::Core::Sectors::Chunk`, 26
 - `uNature::Core::Sectors::Sector`, 87
 - `uNature::Core::Sectors::TICChunk`, 98
 - `uNature::Core::Sectors::UNTerrainSector`, 157
 - `uNature::Core::Targets::UNTarget`, 145
 - `uNature::Core::Terrains::UNTerrain`, 149
- `Backup`
 - `uNature::Core::Terrains::UNTerrainData`, 154
- `bufferedData`
 - `uNature::Core::Networking::UNNetworkManager`, 131
- `canHarvestCollider`
 - `uNature::Core::Pooling::HarvestableTIPoolItem`, 69
- `canModify`
 - `uNature::Core::Pooling::TerrainPoolItem`, 93
- `canRestore`
 - `uNature::Core::Pooling::TerrainPoolItem`, 93
- `Check`
 - `uNature::Core::Targets::UNTarget`, 145
 - `uNature::Core::Terrains::UNTerrain`, 149
- `CheckChunkInBounds`
 - `uNature::Core::FoliageClasses::FoliageCore_↵`
MainManager, 33
- `CheckForMissings`
 - `uNature::Core::Terrains::UNTTreePrototype`, 163
- `CheckForNearbyTreeInstances`
 - `uNature::Core::Sectors::TICChunk`, 98
- `CheckForStreamingBufferedUpdates`
 - `uNature::Core::Networking::UNNetworkManager`, 129
- `CheckForTreeInstancesRespawns`
 - `uNature::Core::Terrains::UNTerrain`, 150
- `CheckForTreePrototypesChange`
 - `uNature::Core::Terrains::UNTerrainData`, 154
- `CheckSplatPaint`
 - `uNature::Core::Utility::UNBrushUtility`, 110
- `CheckTargets`
 - `uNature::Core::Targets::UNTarget`, 145
- `CheckTargetsOnMove`
 - `uNature::Core::Seekers::UNSeeker`, 137
- `ClampNegativeIntoPositive`
 - `uNature::Core::Utility::FoliageWorldMaps`, 61
- `collidersPoolItemInstanceIncrease`
 - `uNature::Core::Terrains::UNTerrain`, 152

- Contains
 - uNature::Core::Collections::UNList, [121](#)
 - uNature::Core::Sectors::Chunk, [26](#)
- ContainsKey
 - uNature::Core::Collections::UNDimensionalList, [112](#)
- ContainsValue
 - uNature::Core::Collections::UNDimensionalList, [112](#)
- ConvertTreeInstanceOnTerrain
 - uNature::Core::Pooling::TerrainPoolItem, [92](#)
- Count
 - uNature::Core::Collections::UNDimensionalList, [113](#)
 - uNature::Core::Collections::UNList, [121](#)
- CreateChunk< T >
 - uNature::Core::Sectors::Chunk, [26](#)
- CreateFoliageInstances
 - uNature::Core::FoliageClasses::FoliageMesh↔ Instance, [50](#)
- CreateNewMesh
 - uNature::Core::FoliageClasses::FoliageMesh↔ Manager, [52](#)
- CreatePool
 - uNature::Core::Pooling::Pool, [76](#)
 - uNature::Core::Targets::UNTarget, [145](#)
 - uNature::Core::Terrains::UNTerrain, [150](#)
- CreatePrototype
 - uNature::Core::FoliageClasses::FoliagePrototype, [57](#)
- DelayActionFrames
 - uNature::Core::Threading::UNThreadManager, [160](#)
- DelayActionSeconds
 - uNature::Core::Threading::UNThreadManager, [160](#)
- DeleteBackup
 - uNature::Core::Terrains::UNTerrainData, [154](#)
- Deserialize
 - uNature::Core::Networking::UNNetworkData, [125](#)
- DestroyManager
 - uNature::Core::FoliageClasses::FoliageCore_↔ MainManager, [34](#)
- DestroyMeshInstance
 - uNature::Core::FoliageClasses::FoliageMesh↔ Manager, [52](#)
- DestroyMeshInstances
 - uNature::Core::FoliageClasses::FoliageMesh↔ Manager, [53](#)
- Dispose
 - uNature::Core::Utility::FoliageGrassMap, [42](#)
- distanceOffset
 - uNature::Core::Terrains::UNTerrain, [152](#)
- DrawBrush
 - uNature::Core::Utility::UNBrushUtility, [110](#)
- DrawGUI
 - uNature::Core::Settings::UNSetting, [139](#)
- DrawShape
 - uNature::Core::UNPhysicsObject, [134](#)
- Equals
 - uNature::Core::Networking::UNNetworkData, [125](#)
 - uNature::Core::Terrains::UNTTreePrototype, [163](#)
- Featured
 - uNature::Core::Extensions::UNExtension, [117](#)
- FetchTreeInstances
 - uNature::Core::Sectors::UNTerrainSector, [158](#)
- fileName
 - uNature::Core::Settings::UNSettings, [142](#)
- FixPosition
 - uNature::Core::Targets::UNTarget, [145](#)
 - uNature::Core::Terrains::UNTerrain, [150](#)
- ForceMapsRestore
 - uNature::Core::FoliageClasses::FoliageManager↔ Instance, [46](#)
- GenerateFoliageMeshInstanceForIndex
 - uNature::Core::FoliageClasses::FoliageMesh↔ Manager, [53](#)
- GenerateFoliageMeshInstances
 - uNature::Core::FoliageClasses::FoliageMesh↔ Manager, [53](#)
- GenerateSector
 - uNature::Core::Terrains::UNTerrain, [150](#)
- GenerateSector< T, T1 >
 - uNature::Core::Sectors::Sector, [87](#)
- GenerateTreeInstances
 - uNature::Core::Sectors::TICChunk, [98](#)
- getChunk
 - uNature::Core::Sectors::Sector, [87](#), [88](#)
- GetChunkID
 - uNature::Core::FoliageClasses::FoliageCore_↔ MainManager, [34](#)
- getChunks
 - uNature::Core::Sectors::Sector, [88](#), [89](#)
- GetDensity
 - uNature::Core::FoliageClasses::FoliageMesh↔ Instance, [50](#)
 - uNature::Core::Utility::FoliageGrassMap, [42](#)
- GetDetailLayer
 - uNature::Core::FoliageClasses::FoliageCore_↔ MainManager, [34](#)
- GetHashCode
 - uNature::Core::Networking::UNNetworkData, [125](#)
 - uNature::Core::Terrains::UNTTreePrototype, [163](#)
- GetHeight
 - uNature::Core::Utility::FoliageWorldMaps, [62](#)
- GetInstance
 - uNature::Core::Terrains::UNTerrainData, [154](#)
- GetLogo
 - uNature::Core::Extensions::UNExtension, [115](#)
- GetOrCreateFoliageManagerInstance
 - uNature::Core::FoliageClasses::FoliageCore_↔ Chunk, [30](#)
- GetPoolOfItem
 - uNature::Core::Pooling::Pool, [76](#)

- GetPreview
 - uNature::Core::FoliageClasses::FoliagePrototype, [57](#)
 - uNature::Core::Terrains::UNTreePrototype, [163](#)
 - uNature::Core::Utility::UN_TerrainTexturePrototype, [107](#)
- GetPrototype
 - uNature::Core::Terrains::UNTerrainData, [155](#)
- HandleColliderDeath
 - uNature::Core::Pooling::HarvestableTIPoolItem, [68](#)
- HandleHealthChange
 - uNature::Core::Pooling::HarvestableTIPoolItem, [68](#)
- HandleTreeInstanceDeath
 - uNature::Core::Pooling::HarvestableTIPoolItem, [68](#)
- HarvestChecks
 - uNature::Core::Seekers::UNSeeker, [137](#)
- heights
 - uNature::Core::Terrains::UNTerrainData, [155](#)
- HelperMethods
 - uNature::Core::Extensions::UNExtension, [116](#)
- Hit
 - uNature::Core::Pooling::HarvestableTIPoolItem, [68](#)
- InBounds
 - uNature::Core::FoliageClasses::FoliageCore_↔
Chunk, [31](#)
- InDistance
 - uNature::Core::Targets::UNTarget, [146](#)
 - uNature::Core::Terrains::UNTerrain, [151](#)
- Initialize
 - uNature::Core::Terrains::UNTerrainData, [155](#)
- InitializeAndCreateIfNotFound
 - uNature::Core::FoliageClasses::FoliageCore_↔
MainManager, [34](#)
- InsertFoliageFromTerrain
 - uNature::Core::FoliageClasses::FoliageCore_↔
MainManager, [34](#)
- instance
 - uNature::Core::FoliageClasses::FoliageDB, [40](#)
- InverseCord
 - uNature::Core::FoliageClasses::FoliageManager↔
Instance, [46](#)
- InverseCordCustom
 - uNature::Core::FoliageClasses::FoliageManager↔
Instance, [46](#)
- InverseCordCustomFloat
 - uNature::Core::FoliageClasses::FoliageManager↔
Instance, [46](#)
- InverseCordFloat
 - uNature::Core::FoliageClasses::FoliageManager↔
Instance, [46](#)
- isActive
 - uNature::Core::Extensions::UNExtension, [117](#)
- IsAlreadyPooled
 - uNature::Core::Pooling::Pool, [76](#)
- isAuth
 - uNature::Core::Networking::UNNetworkManager, [131](#)
- isCollider
 - uNature::Core::Pooling::TerrainPoolItem, [93](#)
- IsDefault
 - uNature::Core::Extensions::UNExtension, [117](#)
- isEnabled
 - uNature::Core::Terrains::UNTreePrototype, [164](#)
- isFoliageInstanceAttached
 - uNature::Core::FoliageClasses::FoliageCore_↔
Chunk, [31](#)
- isMissing
 - uNature::Core::Terrains::UNTreePrototype, [164](#)
- isServer
 - uNature::Core::Networking::UNNetworkManager, [131](#)
- isViewed
 - uNature::Core::Extensions::UNExtension, [116](#)
- itemID_Offset
 - uNature::Core::Pooling::PoolItem, [81](#)
- itemID
 - uNature::Core::Pooling::PoolItem, [82](#)
- items
 - uNature::Core::Pooling::Pool, [78](#)
- LODMeshInstances
 - uNature::Core::FoliageClasses::GPUMesh, [65](#)
- lastMovement
 - uNature::Core::Seekers::UNSeeker, [137](#)
- LoadMethods
 - uNature::Core::Extensions::UNExtension, [116](#)
- locked
 - uNature::Core::Pooling::PoolItem, [81](#)
- Log
 - uNature::Core::Settings::UNSettings, [141](#)
- manageGrass
 - uNature::Core::Terrains::UNTerrain, [152](#)
- manageTrees
 - uNature::Core::Terrains::UNTerrain, [152](#)
- manager
 - uNature::Core::Networking::UNNetworkManager, [131](#)
- MarkDensitiesDirty
 - uNature::Core::FoliageClasses::FoliageMesh↔
Instance, [50](#)
 - uNature::Core::Utility::FoliageGrassMap, [42](#)
- maxHealth
 - uNature::Core::Pooling::HarvestableTIPoolItem, [69](#)
- minHealth
 - uNature::Core::Pooling::HarvestableTIPoolItem, [69](#)
- MoveItem
 - uNature::Core::Pooling::PoolItem, [80](#)
 - uNature::Core::Pooling::TerrainPoolItem, [92](#)
- NormalizeHeight
 - uNature::Core::Utility::FoliageWorldMaps, [62](#)
- OnChunkCreated
 - uNature::Core::FoliageClasses::FoliageCore_↔
Sector, [37](#)

- uNature::Core::FoliageClasses::FoliageSector, 60
- uNature::Core::Sectors::Sector, 89
- uNature::Core::Sectors::UNTerrainSector, 158
- OnClientConnected
 - uNature::Core::Networking::UNNetworkManager, 129
- OnCreated
 - uNature::Core::FoliageClasses::FoliageChunk, 29
 - uNature::Core::FoliageClasses::FoliageCore_↔
Chunk, 31
 - uNature::Core::Pooling::PoolItem, 81
 - uNature::Core::Sectors::Chunk, 27
 - uNature::Core::Sectors::Sector, 89
 - uNature::Core::Sectors::TICChunk, 99
 - uNature::Core::Sectors::UNTerrainSector, 158
- OnDisable
 - uNature::Core::FoliageClasses::BaseInteraction, 22
 - uNature::Core::FoliageClasses::FoliageChunk, 29
 - uNature::Core::FoliageClasses::FoliageCore_↔
MainManager, 35
 - uNature::Core::FoliageClasses::FoliageManager_↔
Instance, 47
 - uNature::Core::FoliageClasses::FoliageMesh_↔
Manager, 54
 - uNature::Core::FoliageClasses::FoliageReceiver, 59
 - uNature::Core::FoliageClasses::TouchBending, 100
 - uNature::Core::Pooling::PoolItem, 81
 - uNature::Core::Sectors::Chunk, 27
 - uNature::Core::Targets::UNTTarget, 146
 - uNature::Core::Terrains::UNTerrain, 151
 - uNature::Core::Threading::ThreadItem, 95
- OnDrawGizmos
 - uNature::Core::FoliageClasses::FoliageCore_↔
Chunk, 31
 - uNature::Core::Sectors::Chunk, 27
 - uNature::Core::Sectors::TICChunk, 99
 - uNature::Core::Targets::UNTTarget, 146
 - uNature::Core::UNPhysicsObject, 134
- OnEnable
 - uNature::Core::Pooling::PoolItem, 81
 - uNature::Core::Sectors::Chunk, 27
 - uNature::Core::Targets::UNTTarget, 146
 - uNature::Core::Terrains::UNTerrain, 151
- OnHarvestableTreeInstancePooled
 - uNature::Core::Networking::UNNetworkManager, 129
- OnItemDamaged
 - uNature::Core::Networking::UNNetworkManager, 130
- OnItemDamagedEvent
 - uNature::Core::Pooling::HarvestableTIPoolItem, 69
- OnItemPooledEvent
 - uNature::Core::Pooling::HarvestableTIPoolItem, 69
- OnItemReturnedToPoolEvent
 - uNature::Core::Pooling::HarvestableTIPoolItem, 69
- OnPool
 - uNature::Core::Pooling::HarvestableTIPoolItem, 68
 - uNature::Core::Pooling::PoolItem, 81
- OnPositionChanged
 - uNature::Core::FoliageClasses::BaseInteraction, 22
 - uNature::Core::FoliageClasses::TouchBending, 100
 - uNature::Core::Terrains::UNTerrain, 151
 - uNature::Core::Threading::ThreadItem, 95
- OnResolutionChanged
 - uNature::Core::FoliageClasses::FoliageCore_↔
Sector, 38
 - uNature::Core::Sectors::Sector, 89
- OnReturnedToPool
 - uNature::Core::Pooling::HarvestableTIPoolItem, 68
 - uNature::Core::Pooling::PoolItem, 81
- OnSizeChanged
 - uNature::Core::FoliageClasses::FoliageChunk, 29
 - uNature::Core::FoliageClasses::FoliageCore_↔
Chunk, 31
 - uNature::Core::Sectors::Chunk, 27
 - uNature::Core::Sectors::TICChunk, 99
- OnStartCreatingChunks
 - uNature::Core::FoliageClasses::FoliageCore_↔
Sector, 38
 - uNature::Core::FoliageClasses::FoliageSector, 60
 - uNature::Core::Sectors::Sector, 89
 - uNature::Core::Sectors::UNTerrainSector, 158
- OnTerrainChanged
 - uNature::Core::Terrains::UNTerrain, 151
- OnThreadProcess
 - uNature::Core::Threading::UNThreadManager, 160
- OpenAssetStore
 - uNature::Core::Extensions::UNExtension, 116
- OpenDocs
 - uNature::Core::Extensions::UNExtension, 116
- owner
 - uNature::Core::Pooling::Pool, 78
- Pack< T1, T2 >
 - uNature::Core::Networking::UNNetworkData, 125
- Pool
 - uNature::Core::Pooling::PoolItem, 82
 - uNature::Core::Targets::UNTTarget, 147
- PoolAmount
 - uNature::Core::Targets::UNTTarget, 147
- PoolItem
 - uNature::Core::Pooling::Pool, 76
- PoolTypeSerializedName
 - uNature::Core::Targets::UNTTarget, 147
- PoolTypes
 - uNature::Core::Pooling::PoolItem, 82
- ProjectName
 - uNature::Core::Settings::UNSettings, 142
- ProjectPath
 - uNature::Core::Settings::UNSettings, 142
- ProjectVersion

- uNature::Core::Settings::UNSettings, 142
- prototypeObject
 - uNature::Core::Terrains::UNTreePrototype, 164
- PublisherName
 - uNature::Core::Extensions::UNExtension, 117
- Raycast
 - uNature::Core::UNPhysicsObject, 134
- raycastDistance
 - uNature::Core::Seekers::UNSeeker, 138
- raycastMask
 - uNature::Core::Seekers::UNSeeker, 138
- ReGenerateGlobally
 - uNature::Core::Utility::FoliageWorldMaps, 62
- realItemID
 - uNature::Core::Pooling::PoolItem, 82
- RegenerateQueueInstances
 - uNature::Core::FoliageClasses::FoliageMesh↔Manager, 54
- Remove
 - uNature::Core::Collections::UNList, 121
- RemoveDuplications
 - uNature::Core::Pooling::Pool, 77
- RemoveFromPool
 - uNature::Core::Pooling::Pool, 77
- RemoveGrassMap
 - uNature::Core::FoliageClasses::FoliageCore↔MainManager, 35
- RemovePrototype
 - uNature::Core::FoliageClasses::FoliageDB, 39
- RemoveTreeInstanceFromTerrain
 - uNature::Core::Pooling::TerrainPoolItem, 93
- removedTreeInstanceHeight
 - uNature::Core::Terrains::UNTerrain, 152
- ResetChunk
 - uNature::Core::Sectors::Chunk, 27
 - uNature::Core::Sectors::TICheck, 99
- ResetChunks
 - uNature::Core::Sectors::Sector, 90
- ResetDefaults
 - uNature::Core::Settings::UNSettings, 142
- ResetFarAway
 - uNature::Core::Pooling::Pool, 77
- ResetGrassMap
 - uNature::Core::FoliageClasses::FoliageCore↔MainManager, 35
- Resize
 - uNature::Core::Utility::UNBrushUtility, 111
- RestoreChanges
 - uNature::Core::Utility::FoliageWorldMaps, 62
- RestoreTreeInstanceToTerrain
 - uNature::Core::Pooling::TerrainPoolItem, 93
- ReturnToPool
 - uNature::Core::Pooling::Pool, 77
- RunOnThread
 - uNature::Core::Threading::UNThreadManager, 160
- RunOnUnityThread
 - uNature::Core::Threading::UNThreadManager, 160
- SafeGetMap
 - uNature::Core::Utility::FoliageGrassMap, 42
- Save
 - uNature::Core::Utility::FoliageNormalMap, 55
 - uNature::Core::Utility::FoliageWorldMaps, 62
- SaveAllMaps
 - uNature::Core::Utility::FoliageWorldMaps, 62
- SaveDelayedMaps
 - uNature::Core::FoliageClasses::FoliageCore↔MainManager, 35
- sectorResolution
 - uNature::Core::Terrains::UNTerrain, 152
- seekingDistance
 - uNature::Core::Seekers::UNSeeker, 138
- SendEvent
 - uNature::Core::Networking::UNNetworkManager, 130
- SendToClients
 - uNature::Core::Networking::UNNetworkData, 125
 - uNature::Core::Networking::UNNetworkManager, 130
- SendToConnection
 - uNature::Core::Networking::UNNetworkData, 126
 - uNature::Core::Networking::UNNetworkManager, 130
- SendToOthers
 - uNature::Core::Networking::UNNetworkData, 127
 - uNature::Core::Networking::UNNetworkManager, 131
- SendToServer
 - uNature::Core::Networking::UNNetworkData, 127
 - uNature::Core::Networking::UNNetworkManager, 131
- SendUpdateEventToLinkedTerrains
 - uNature::Core::Terrains::UNTerrainData, 155
- Serialize
 - uNature::Core::Networking::UNNetworkData, 127
- SetDensity
 - uNature::Core::Utility::FoliageGrassMap, 42
- SetDensityFast
 - uNature::Core::Utility::FoliageGrassMap, 43
- SetDetailLayer
 - uNature::Core::FoliageClasses::FoliageCore↔MainManager, 35
- SetPixels32Delayed
 - uNature::Core::Utility::FoliageWorldMaps, 63
- Start
 - uNature::Core::Seekers::UNSeeker, 137
- terrainData
 - uNature::Core::Terrains::UNTerrainData, 155
- terrains
 - uNature::Core::Terrains::UNTerrain, 152
- this[int index]
 - uNature::Core::Collections::UNDimensionalList, 113

- uNature::Core::Collections::UNList, 121
- threadEnabled
 - uNature::Core::Threading::UNThreadManager, 161
- threadWorkersCount
 - uNature::Core::Threading::UNThreadManager, 161
- TransformCord
 - uNature::Core::FoliageClasses::FoliageManager↔ Instance, 47
- TransformCordCustom
 - uNature::Core::FoliageClasses::FoliageManager↔ Instance, 47
- TransformCordCustomFloat
 - uNature::Core::FoliageClasses::FoliageManager↔ Instance, 47
- TransformCordFloat
 - uNature::Core::FoliageClasses::FoliageManager↔ Instance, 48
- TransformNormals
 - uNature::Core::Utility::FoliageWorldMaps, 63
- TryAddKey
 - uNature::Core::Collections::UNDimensionalList, 113
- TryGet
 - uNature::Core::Collections::UNList, 121
- TryGetType< T >
 - uNature::Core::Pooling::Pool, 77
- TryPool< T >
 - uNature::Core::Pooling::Pool, 78
- TryResetOnUID
 - uNature::Core::Pooling::Pool, 78
- UNNetworkManager
 - uNature::Core::Networking::UNNetworkManager, 129
- UNSettingCategories
 - uNature::Core::Settings, 18
- UNTreePrototype
 - uNature::Core::Terrains::UNTreePrototype, 162
- uNature, 13
- uNature.Core, 13
- uNature.Core.ClassExtensions, 13
- uNature.Core.Collections, 14
- uNature.Core.Collections.UNDimensionalList< T >, 112
- uNature.Core.Collections.UNList< T >, 120
- uNature.Core.Editor, 14
- uNature.Core.Editor.Helpers, 14
- uNature.Core.Editor.Helpers.UNEditorHelpers, 113
- uNature.Core.Extensions, 14
- uNature.Core.Extensions.MethodHelperAttribute, 72
- uNature.Core.Extensions.UN_ForgeNetworking, 102
- uNature.Core.Extensions.UN_GAIA, 102
- uNature.Core.Extensions.UN_GENA, 103
- uNature.Core.Extensions.UN_MapMagic, 104
- uNature.Core.Extensions.UN_PhotonBolt, 105
- uNature.Core.Extensions.UN_PhotonCloud, 105
- uNature.Core.Extensions.UN_TerrainComposer, 106
- uNature.Core.Extensions.UN_UFPS, 107
- uNature.Core.Extensions.UN_UNet, 108
- uNature.Core.Extensions.UN_WorldStreamer, 109
- uNature.Core.Extensions.UNExtension, 114
- uNature.Core.Extensions.UNExtensionsEditor, 118
- uNature.Core.FoliageClasses, 15
- uNature.Core.FoliageClasses.BaseInteraction, 21
- uNature.Core.FoliageClasses.FoliageChunk, 28
- uNature.Core.FoliageClasses.FoliageCore_Chunk, 29
- uNature.Core.FoliageClasses.FoliageCore_Main↔ Manager, 32
- uNature.Core.FoliageClasses.FoliageCore_Sector, 37
- uNature.Core.FoliageClasses.FoliageDB, 38
- uNature.Core.FoliageClasses.FoliageDynamicSurface, 40
- uNature.Core.FoliageClasses.FoliageLODLevel, 44
- uNature.Core.FoliageClasses.FoliageManagerInstance, 44
- uNature.Core.FoliageClasses.FoliageMesh, 48
- uNature.Core.FoliageClasses.FoliageMeshInstance, 49
- uNature.Core.FoliageClasses.FoliageMeshInstances↔ Group, 50
- uNature.Core.FoliageClasses.FoliageMeshManager, 51
- uNature.Core.FoliageClasses.FoliagePrototype, 55
- uNature.Core.FoliageClasses.FoliageReceiver, 58
- uNature.Core.FoliageClasses.FoliageSector, 59
- uNature.Core.FoliageClasses.GPUMesh, 64
- uNature.Core.FoliageClasses.GPUMeshLOD, 65
- uNature.Core.FoliageClasses.PaintBrush, 73
- uNature.Core.FoliageClasses.RenderingQueue, 83
- uNature.Core.FoliageClasses.RenderingQueue↔ Instance, 83
- uNature.Core.FoliageClasses.RenderingQueueMesh↔ InstanceSimulator, 84
- uNature.Core.FoliageClasses.RenderingQueue↔ Receiver, 85
- uNature.Core.FoliageClasses.TouchBending, 99
- uNature.Core.FoliageClasses.TouchBendingEditor, 101
- uNature.Core.FoliageClasses.UNFoliageEditor, 119
- uNature.Core.FoliageClasses.UNFoliageManager↔ Editor, 119
- uNature.Core.FoliageClasses.UNMeshData, 123
- uNature.Core.FoliageClasses.WindSettings, 165
- uNature.Core.FoliageClasses.WindZones, 165
- uNature.Core.IUTCPysicsIgnored, 71
- uNature.Core.Math, 16
- uNature.Core.Networking, 16
- uNature.Core.Networking.BaseUNNetworkData, 23
- uNature.Core.Networking.UNNetworkData< T >, 123
- uNature.Core.Networking.UNNetworkManager< T1, T2 >, 127
- uNature.Core.Networking.UNNetworkPlayerController, 132
- uNature.Core.Pooling, 16
- uNature.Core.Pooling.HarvestableTIPoolItem, 66
- uNature.Core.Pooling.IHarvestableItem, 70
- uNature.Core.Pooling.IPoolComponent, 71
- uNature.Core.Pooling.Pool, 74

- uNature.Core.Pooling.PoolItem, 79
- uNature.Core.Pooling.TerrainPoolItem, 91
- uNature.Core.Sectors, 17
- uNature.Core.Sectors.Chunk, 24
- uNature.Core.Sectors.ChunkObject, 28
- uNature.Core.Sectors.GrassLODLevel, 65
- uNature.Core.Sectors.Sector, 85
- uNature.Core.Sectors.TIChunk, 97
- uNature.Core.Sectors.TreeFetchingTask_MultiThreaded, 101
- uNature.Core.Sectors.UNTerrainSector, 156
- uNature.Core.Seekers, 17
- uNature.Core.Seekers.UNSeeker, 136
- uNature.Core.Seekers.UNSeekerEditor, 138
- uNature.Core.Settings, 18
- uNature.Core.Settings.UNSetting, 138
- uNature.Core.Settings.UNSettingAttribute, 139
- uNature.Core.Settings.UNSettingCategory, 140
- uNature.Core.Settings.UNSettings, 141
- uNature.Core.Settings.UNSettingsEditor, 142
- uNature.Core.Targets, 18
- uNature.Core.Targets.UNTarget, 143
- uNature.Core.Targets.UNTargetEditor, 147
- uNature.Core.Terrains, 18
- uNature.Core.Terrains.UNTerrain, 148
- uNature.Core.Terrains.UNTerrainData, 153
- uNature.Core.Terrains.UNTerrainEditor, 156
- uNature.Core.Terrains.UNTreePrototype, 161
- uNature.Core.Threading, 19
- uNature.Core.Threading.IThreadTask, 71
- uNature.Core.Threading.ThreadItem, 94
- uNature.Core.Threading.ThreadTask, 95–97
- uNature.Core.Threading.UNThreadManager, 159
- uNature.Core.UNPhysicsHit_Grass, 132
- uNature.Core.UNPhysicsHitsArray, 133
- uNature.Core.UNPhysicsObject, 133
- uNature.Core.UNPhysicsTemplate, 135
- uNature.Core.Utility, 19
- uNature.Core.Utility.BasePrototypeItem, 23
- uNature.Core.Utility.FoliageGrassMap, 41
- uNature.Core.Utility.FoliageHeightMap, 43
- uNature.Core.Utility.FoliageNormalMap, 54
- uNature.Core.Utility.FoliageWorldMaps, 60
- uNature.Core.Utility.UN_TerrainTexturePrototype, 106
- uNature.Core.Utility.UNBrushUtility, 109
- uNature.Core.Utility.UNCombineInstance, 111
- uNature.Core.Utility.UNDictionary< T, T1 >, 111
- uNature.Core.Utility.UNFastList< T >, 118
- uNature.Core.Utility.UNMap, 122
- uNature.Core.Utility.Vector3_XZ_FAST, 165
- uNature.Core.Vector2i, 164
- uNature.Demo, 20
- uNature.Demo.UN_FirstPersonController, 101
- uNature.Demo.UN_MouseLook, 104
- uNature.Wrappers, 20
- uNature.Wrappers.Linq, 20
- uNature.Wrappers.Linq.IOrderedEnumerable< T↔ Element >, 70
- uNature.Wrappers.Linq.OrderedEnumerable< T↔ Element >, 72
- uNature.Wrappers.Linq.OrderedSequence< TElement, TKey >, 73
- uNature.Wrappers.Linq.QuickSort< TElement >, 82
- uNature.Wrappers.Linq.SortContext< TElement >, 90
- uNature.Wrappers.Linq.SortSequenceContext< T↔ Element, TKey >, 91
- uNature::Core::Collections::UNDimensionalList
 - ContainsKey, 112
 - ContainsValue, 112
 - Count, 113
 - this[int index], 113
 - TryAddKey, 113
- uNature::Core::Collections::UNList
 - Add, 120
 - Contains, 121
 - Count, 121
 - Remove, 121
 - this[int index], 121
 - TryGet, 121
- uNature::Core::Extensions::UNExtension
 - AssetDescription, 116
 - AssetDocumentationName, 116
 - AssetLogoName, 117
 - AssetName, 117
 - AssetNameSpace, 117
 - AssetStoreAdress, 117
 - Featured, 117
 - GetLogo, 115
 - HelperMethods, 116
 - isActive, 117
 - IsDefault, 117
 - isViewed, 116
 - LoadMethods, 116
 - OpenAssetStore, 116
 - OpenDocs, 116
 - PublisherName, 117
- uNature::Core::FoliageClasses::BaseInteraction
 - OnDisable, 22
 - OnPositionChanged, 22
 - UpdateInteraction, 22
- uNature::Core::FoliageClasses::FoliageChunk
 - OnCreated, 29
 - OnDisable, 29
 - OnSizeChanged, 29
- uNature::Core::FoliageClasses::FoliageCore_Chunk
 - Awake, 30
 - GetOrCreateFoliageManagerInstance, 30
 - InBounds, 31
 - isFoliageInstanceAttached, 31
 - OnCreated, 31
 - OnDrawGizmos, 31
 - OnSizeChanged, 31
- uNature::Core::FoliageClasses::FoliageCore_Main↔ Manager
 - Awake, 33
 - CheckChunkInBounds, 33

- DestroyManager, [34](#)
- GetChunkID, [34](#)
- GetDetailLayer, [34](#)
- InitializeAndCreatelfNotFound, [34](#)
- InsertFoliageFromTerrain, [34](#)
- OnDisable, [35](#)
- RemoveGrassMap, [35](#)
- ResetGrassMap, [35](#)
- SaveDelayedMaps, [35](#)
- SetDetailLayer, [35](#)
- Update, [36](#)
- UpdateHeights, [36](#)
- UpdateHeightsOnTerrain, [36](#)
- WarmUpGrassMaps, [36](#)
- uNature::Core::FoliageClasses::FoliageCore_Sector
 - OnChunkCreated, [37](#)
 - OnResolutionChanged, [38](#)
 - OnStartCreatingChunks, [38](#)
- uNature::Core::FoliageClasses::FoliageDB
 - AddPrototype, [39](#)
 - instance, [40](#)
 - RemovePrototype, [39](#)
 - UpdateShaderGeneralSettings, [39](#)
 - UpdateShaderWindSettings, [39](#)
- uNature::Core::FoliageClasses::FoliageManager↔
 - Instance
 - ForceMapsRestore, [46](#)
 - InverseCord, [46](#)
 - InverseCordCustom, [46](#)
 - InverseCordCustomFloat, [46](#)
 - InverseCordFloat, [46](#)
 - OnDisable, [47](#)
 - TransformCord, [47](#)
 - TransformCordCustom, [47](#)
 - TransformCordCustomFloat, [47](#)
 - TransformCordFloat, [48](#)
- uNature::Core::FoliageClasses::FoliageMeshInstance
 - CreateFoliageInstances, [50](#)
 - GetDensity, [50](#)
 - MarkDensitiesDirty, [50](#)
- uNature::Core::FoliageClasses::FoliageMeshManager
 - CreateNewMesh, [52](#)
 - DestroyMeshInstance, [52](#)
 - DestroyMeshInstances, [53](#)
 - GenerateFoliageMeshInstanceForIndex, [53](#)
 - GenerateFoliageMeshInstances, [53](#)
 - OnDisable, [54](#)
 - RegenerateQueueInstances, [54](#)
 - Update, [54](#)
 - UpdateMeshBounds, [54](#)
- uNature::Core::FoliageClasses::FoliagePrototype
 - ApplyColorMap, [57](#)
 - ApplyGrassMap, [57](#)
 - ApplyWind, [57](#)
 - CreatePrototype, [57](#)
 - GetPreview, [57](#)
 - UpdateManagerInformation, [57](#)
- uNature::Core::FoliageClasses::FoliageReceiver
 - OnDisable, [59](#)
 - Update, [59](#)
- uNature::Core::FoliageClasses::FoliageSector
 - OnChunkCreated, [60](#)
 - OnStartCreatingChunks, [60](#)
- uNature::Core::FoliageClasses::GPUMesh
 - LODMeshInstances, [65](#)
- uNature::Core::FoliageClasses::TouchBending
 - OnDisable, [100](#)
 - OnPositionChanged, [100](#)
- uNature::Core::Networking::BaseUNNetworkData
 - UnPack, [24](#)
- uNature::Core::Networking::UNNetworkData
 - Deserialize, [125](#)
 - Equals, [125](#)
 - GetHashCode, [125](#)
 - Pack< T1, T2 >, [125](#)
 - SendToClients, [125](#)
 - SendToConnection, [126](#)
 - SendToOthers, [127](#)
 - SendToServer, [127](#)
 - Serialize, [127](#)
 - UnPack, [127](#)
- uNature::Core::Networking::UNNetworkManager
 - Awake, [129](#)
 - bufferedData, [131](#)
 - CheckForStreamingBufferedUpdates, [129](#)
 - isAuth, [131](#)
 - isServer, [131](#)
 - manager, [131](#)
 - OnClientConnected, [129](#)
 - OnHarvestableTreeInstancePooled, [129](#)
 - OnItemDamaged, [130](#)
 - SendEvent, [130](#)
 - SendToClients, [130](#)
 - SendToConnection, [130](#)
 - SendToOthers, [131](#)
 - SendToServer, [131](#)
 - UNNetworkManager, [129](#)
 - UpdatePermissions, [131](#)
- uNature::Core::Pooling::HarvestableTIPoolItem
 - Awake, [68](#)
 - canHarvestCollider, [69](#)
 - HandleColliderDeath, [68](#)
 - HandleHealthChange, [68](#)
 - HandleTreeInstanceDeath, [68](#)
 - Hit, [68](#)
 - maxHealth, [69](#)
 - minHealth, [69](#)
 - OnItemDamagedEvent, [69](#)
 - OnItemPooledEvent, [69](#)
 - OnItemReturnedToPoolEvent, [69](#)
 - OnPool, [68](#)
 - OnReturnedToPool, [68](#)
- uNature::Core::Pooling::Pool
 - AddToPool, [75](#)
 - CreatePool, [76](#)
 - GetPoolOfItem, [76](#)

- IsAlreadyPooled, 76
- items, 78
- owner, 78
- PoolItem, 76
- RemoveDuplications, 77
- RemoveFromPool, 77
- ResetFarAway, 77
- ReturnToPool, 77
- TryGetType< T >, 77
- TryPool< T >, 78
- TryResetOnUID, 78
- uNature::Core::Pooling::PoolItem
 - _gameObject, 81
 - Awake, 80
 - itemID_Offset, 81
 - itemID, 82
 - locked, 81
 - MovelItem, 80
 - OnCreated, 81
 - OnDisable, 81
 - OnEnable, 81
 - OnPool, 81
 - OnReturnedToPool, 81
 - Pool, 82
 - PoolTypes, 82
 - realItemID, 82
 - uid, 82
 - used, 82
- uNature::Core::Pooling::TerrainPoolItem
 - _terrain, 93
 - canModify, 93
 - canRestore, 93
 - ConvertTreeInstanceOnTerrain, 92
 - isCollider, 93
 - MovelItem, 92
 - RemoveTreeInstanceFromTerrain, 93
 - RestoreTreeInstanceToTerrain, 93
- uNature::Core::Sectors::Chunk
 - Awake, 26
 - Contains, 26
 - CreateChunk< T >, 26
 - OnCreated, 27
 - OnDisable, 27
 - OnDrawGizmos, 27
 - OnEnable, 27
 - OnSizeChanged, 27
 - ResetChunk, 27
- uNature::Core::Sectors::Sector
 - ApplicationQuit, 87
 - Awake, 87
 - GenerateSector< T, T1 >, 87
 - getChunk, 87, 88
 - getChunks, 88, 89
 - OnChunkCreated, 89
 - OnCreated, 89
 - OnResolutionChanged, 89
 - OnStartCreatingChunks, 89
 - ResetChunks, 90
- uNature::Core::Sectors::TICChunk
 - AddTreeInstance, 98
 - Awake, 98
 - CheckForNearbyTreeInstances, 98
 - GenerateTreeInstances, 98
 - OnCreated, 99
 - OnDrawGizmos, 99
 - OnSizeChanged, 99
 - ResetChunk, 99
- uNature::Core::Sectors::UNTerrainSector
 - ApplicationQuit, 157
 - Awake, 157
 - FetchTreeInstances, 158
 - OnChunkCreated, 158
 - OnCreated, 158
 - OnStartCreatingChunks, 158
- uNature::Core::Seekers::UNSeeker
 - _treesCheckDistance, 137
 - attackTrees, 137
 - CheckTargetsOnMove, 137
 - HarvestChecks, 137
 - lastMovement, 137
 - raycastDistance, 138
 - raycastMask, 138
 - seekingDistance, 138
 - Start, 137
 - Update, 137
- uNature::Core::Settings
 - UNSettingCategories, 18
- uNature::Core::Settings::UNSetting
 - DrawGUI, 139
- uNature::Core::Settings::UNSettings
 - fileName, 142
 - Log, 141
 - ProjectName, 142
 - ProjectPath, 142
 - ProjectVersion, 142
 - ResetDefaults, 142
- uNature::Core::Targets::UNTTarget
 - Awake, 145
 - Check, 145
 - CheckTargets, 145
 - CreatePool, 145
 - FixPosition, 145
 - InDistance, 146
 - OnDisable, 146
 - OnDrawGizmos, 146
 - OnEnable, 146
 - Pool, 147
 - PoolAmount, 147
 - PoolTypeSerializedName, 147
 - Update, 146
 - useMultithreadedCheck, 147
 - worldTargets, 147
- uNature::Core::Terrains::UNTerrain
 - Awake, 149
 - Check, 149
 - CheckForTreeInstancesRespawns, 150

- collidersPoolItemInstanceIncrease, 152
- CreatePool, 150
- distanceOffset, 152
- FixPosition, 150
- GenerateSector, 150
- InDistance, 151
- manageGrass, 152
- manageTrees, 152
- OnDisable, 151
- OnEnable, 151
- OnPositionChanged, 151
- OnTerrainChanged, 151
- removedTreeInstanceHeight, 152
- sectorResolution, 152
- terrains, 152
- verifyTreeInstancesChangeRoutine, 152
- uNature::Core::Terrains::UNTerrainData
 - ApplyBackup, 154
 - Backup, 154
 - CheckForTreePrototypesChange, 154
 - DeleteBackup, 154
 - GetInstance, 154
 - GetPrototype, 155
 - heights, 155
 - Initialize, 155
 - SendUpdateEventToLinkedTerrains, 155
 - terrainData, 155
 - UpdateMultithreadedVariables, 155
- uNature::Core::Terrains::UNTreePrototype
 - CheckForMissings, 163
 - Equals, 163
 - GetHashCode, 163
 - GetPreview, 163
 - isEnabled, 164
 - isMissing, 164
 - prototypeObject, 164
 - UNTreePrototype, 162
- uNature::Core::Threading::ThreadItem
 - _threadItems, 95
 - OnDisable, 95
 - OnPositionChanged, 95
 - Update, 95
 - UpdateItem, 95
- uNature::Core::Threading::UNThreadManager
 - DelayActionFrames, 160
 - DelayActionSeconds, 160
 - OnThreadProcess, 160
 - RunOnThread, 160
 - RunOnUnityThread, 160
 - threadEnabled, 161
 - threadWorkersCount, 161
 - UpdateThreadItems, 161
 - updateThreadItemsTime, 161
- uNature::Core::UNPhysicsObject
 - DrawShape, 134
 - OnDrawGizmos, 134
 - Raycast, 134
 - UpdateBounds, 135
- uNature::Core::Utility::FoliageGrassMap
 - Dispose, 42
 - GetDensity, 42
 - MarkDensitiesDirty, 42
 - SafeGetMap, 42
 - SetDensity, 42
 - SetDensityFast, 43
 - UpdateGrassMaps, 43
- uNature::Core::Utility::FoliageNormalMap
 - Save, 55
- uNature::Core::Utility::FoliageWorldMaps
 - ClampNegativeIntoPositive, 61
 - GetHeight, 62
 - NormalizeHeight, 62
 - ReGenerateGlobally, 62
 - RestoreChanges, 62
 - Save, 62
 - SaveAllMaps, 62
 - SetPixels32Delayed, 63
 - TransformNormals, 63
 - UpdateHeightAndNormal, 63
 - UpdateHeightsAndNormals, 64
- uNature::Core::Utility::UN_TerrainTexturePrototype
 - GetPreview, 107
- uNature::Core::Utility::UNBrushUtility
 - CheckSplatPaint, 110
 - DrawBrush, 110
 - Resize, 111
- uid
 - uNature::Core::Pooling::PoolItem, 82
- UnPack
 - uNature::Core::Networking::BaseUNNetworkData, 24
 - uNature::Core::Networking::UNNetworkData, 127
- Update
 - uNature::Core::FoliageClasses::FoliageCore_↔
 - MainManager, 36
 - uNature::Core::FoliageClasses::FoliageMesh↔
 - Manager, 54
 - uNature::Core::FoliageClasses::FoliageReceiver, 59
 - uNature::Core::Seekers::UNSeeker, 137
 - uNature::Core::Targets::UNTarget, 146
 - uNature::Core::Threading::ThreadItem, 95
- UpdateBounds
 - uNature::Core::UNPhysicsObject, 135
- UpdateGrassMaps
 - uNature::Core::Utility::FoliageGrassMap, 43
- UpdateHeightAndNormal
 - uNature::Core::Utility::FoliageWorldMaps, 63
- UpdateHeights
 - uNature::Core::FoliageClasses::FoliageCore_↔
 - MainManager, 36
- UpdateHeightsAndNormals
 - uNature::Core::Utility::FoliageWorldMaps, 64
- UpdateHeightsOnTerrain
 - uNature::Core::FoliageClasses::FoliageCore_↔
 - MainManager, 36

- UpdateInteraction
 - uNature::Core::FoliageClasses::BaseInteraction,
[22](#)
- UpdateItem
 - uNature::Core::Threading::ThreadItem, [95](#)
- UpdateManagerInformation
 - uNature::Core::FoliageClasses::FoliagePrototype,
[57](#)
- UpdateMeshBounds
 - uNature::Core::FoliageClasses::FoliageMesh↔
Manager, [54](#)
- UpdateMultithreadedVariables
 - uNature::Core::Terrains::UNTerrainData, [155](#)
- UpdatePermissions
 - uNature::Core::Networking::UNNetworkManager,
[131](#)
- UpdateShaderGeneralSettings
 - uNature::Core::FoliageClasses::FoliageDB, [39](#)
- UpdateShaderWindSettings
 - uNature::Core::FoliageClasses::FoliageDB, [39](#)
- UpdateThreadItems
 - uNature::Core::Threading::UNThreadManager,
[161](#)
- updateThreadItemsTime
 - uNature::Core::Threading::UNThreadManager,
[161](#)
- useMultithreadedCheck
 - uNature::Core::Targets::UNTarget, [147](#)
- used
 - uNature::Core::Pooling::PoolItem, [82](#)
- verifyTreeInstancesChangeRoutine
 - uNature::Core::Terrains::UNTerrain, [152](#)
- WarmUpGrassMaps
 - uNature::Core::FoliageClasses::FoliageCore↔
MainManager, [36](#)
- worldTargets
 - uNature::Core::Targets::UNTarget, [147](#)