

Skylight

Generated by Doxygen 1.8.7

Thu Jul 17 2014 11:40:43

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	Namespace Documentation	7
4.1	Package Skylight	7
4.2	Package Skylight.Arguments	8
4.3	Package Skylight.Blocks	8
4.4	Package Skylight.Miscellaneous	9
4.5	Package Skylight.Physics	9
4.6	Package Skylight.Properties	9
5	Class Documentation	11
5.1	Skylight.Access Class Reference	11
5.1.1	Event Documentation	11
5.1.1.1	GainAccessEvent	11
5.2	Skylight.Add Class Reference	11
5.3	Skylight.AddSpecialBlock Class Reference	12
5.3.1	Event Documentation	12
5.3.1.1	RotateEvent	12
5.4	Skylight.Autotext Class Reference	12
5.4.1	Event Documentation	12
5.4.1.1	AutotextEvent	12
5.5	Skylight.Miscellaneous.SmileyIds.Beta Class Reference	13
5.5.1	Detailed Description	13
5.5.2	Member Data Documentation	13
5.5.2.1	Tear	13
5.6	Skylight.Blocks.Block Class Reference	13

5.6.1	Detailed Description	14
5.6.2	Constructor & Destructor Documentation	14
5.6.2.1	Block	14
5.6.3	Member Data Documentation	14
5.6.3.1	Right	14
5.6.4	Property Documentation	14
5.6.4.1	Direction	14
5.6.4.2	Id	14
5.6.4.3	IsSolid	14
5.6.4.4	Placer	15
5.6.4.5	X	15
5.6.4.6	Y	15
5.6.4.7	Z	15
5.7	Skylight.BlockChanged Class Reference	15
5.7.1	Event Documentation	15
5.7.1.1	NormalBlockEvent	15
5.8	Skylight.Arguments.BlockEventArgs Class Reference	15
5.8.1	Detailed Description	16
5.8.2	Constructor & Destructor Documentation	16
5.8.2.1	BlockEventArgs	16
5.8.3	Property Documentation	16
5.8.3.1	Origin	16
5.8.3.2	Placed	16
5.8.3.3	Placer	17
5.9	Skylight.Bot Class Reference	17
5.9.1	Detailed Description	18
5.9.2	Member Enumeration Documentation	18
5.9.2.1	AccountType	18
5.9.3	Constructor & Destructor Documentation	18
5.9.3.1	Bot	18
5.9.4	Member Function Documentation	18
5.9.4.1	Disconnect	18
5.9.4.2	Join	19
5.9.4.3	LogIn	20
5.9.5	Property Documentation	20
5.9.5.1	BlockDelay	20
5.9.5.2	ChatPrefix	20
5.9.5.3	Client	20
5.9.5.4	Connection	20
5.9.5.5	CurrentRoom	20

5.9.5.6	IsConnected	20
5.9.5.7	Joined	20
5.9.5.8	Push	20
5.9.5.9	R	20
5.9.5.10	ShouldTick	20
5.9.5.11	SpeechDelay	21
5.10	Skylight.Chat Class Reference	21
5.10.1	Event Documentation	21
5.10.1.1	NormalChatEvent	21
5.11	Skylight.Arguments.ChatEventArgs Class Reference	21
5.11.1	Detailed Description	22
5.11.2	Constructor & Destructor Documentation	22
5.11.2.1	ChatEventArgs	22
5.11.3	Property Documentation	22
5.11.3.1	Message	22
5.11.3.2	Origin	22
5.11.3.3	Speaker	22
5.12	Skylight.ChatOld Class Reference	22
5.12.1	Event Documentation	23
5.12.1.1	SayOldEvent	23
5.13	Skylight.ClearMap Class Reference	23
5.13.1	Event Documentation	23
5.13.1.1	ClearEvent	23
5.14	Skylight.Blocks.CoinBlock Class Reference	23
5.14.1	Detailed Description	24
5.14.2	Constructor & Destructor Documentation	24
5.14.2.1	CoinBlock	24
5.14.3	Property Documentation	24
5.14.3.1	CoinsRequired	24
5.14.3.2	IsGate	24
5.15	Skylight.CoinObject Class Reference	24
5.16	Skylight.Physics.Config Class Reference	25
5.16.1	Detailed Description	28
5.16.2	Member Data Documentation	28
5.16.2.1	AdminColor	28
5.16.2.2	ArmorAuthToken	28
5.16.2.3	ArmorUserid	28
5.16.2.4	CameraLag	28
5.16.2.5	DebugFriendrequest	28
5.16.2.6	DebugInvitation	28

5.16.2.7	DebugNews	28
5.16.2.8	DebugProfile	28
5.16.2.9	DefaultColor	28
5.16.2.10	DefaultColorDark	28
5.16.2.11	DeveloperServer	28
5.16.2.12	DevelopmentModeAutojoinRoom	29
5.16.2.13	DisableCookie	29
5.16.2.14	DisableTracking	29
5.16.2.15	EnableDebugShadow	29
5.16.2.16	ForceArmor	29
5.16.2.17	ForceBeta	29
5.16.2.18	ForceMouseBreaker	29
5.16.2.19	FriendColor	29
5.16.2.20	FriendColorDark	29
5.16.2.21	GuestColor	29
5.16.2.22	Height	29
5.16.2.23	IsMobile	29
5.16.2.24	MaxDailyWoot	30
5.16.2.25	MaxFrameRate	30
5.16.2.26	Maxwidth	30
5.16.2.27	Minwidth	30
5.16.2.28	ModColor	30
5.16.2.29	MousebreakerAuthtoken	30
5.16.2.30	PhysicsBaseDrag	30
5.16.2.31	PhysicsBoost	30
5.16.2.32	PhysicsGravity	30
5.16.2.33	PhysicsJumpHeight	30
5.16.2.34	PhysicsMsPerTick	30
5.16.2.35	PhysicsMudBuoyancy	30
5.16.2.36	PhysicsMudDrag	31
5.16.2.37	PhysicsNoModifierDrag	31
5.16.2.38	PhysicsQueueLength	31
5.16.2.39	PhysicsVariableMultiplier	31
5.16.2.40	PhysicsWaterBuoyancy	31
5.16.2.41	PhysicsWaterDrag	31
5.16.2.42	PlayerioGameld	31
5.16.2.43	RunInDevelopmentMode	31
5.16.2.44	ServerTypeAuthroom	31
5.16.2.45	ServerTypeBetaroom	31
5.16.2.46	ServerTypeBlacklistroom	31

5.16.2.47 ServerTypeGuestserviceroom	31
5.16.2.48 ServerTypeNormalroom	32
5.16.2.49 ServerTypeServiceroom	32
5.16.2.50 ServerTypeTrackingroom	32
5.16.2.51 ServerTypeTutorialroom	32
5.16.2.52 ServerTypeVersion	32
5.16.2.53 ShopPotionMax	32
5.16.2.54 ShowBlacklistInvitation	32
5.16.2.55 ShowDebugFriendrequest	32
5.16.2.56 ShowDebugProfile	32
5.16.2.57 ShowDisabledShopitems	32
5.16.2.58 TutorialNames	32
5.16.2.59 UrlBlog	32
5.16.2.60 UrlClubmemberAboutPage	33
5.16.2.61 UrlHelpPage	33
5.16.2.62 UrlTermsPage	33
5.16.2.63 UseDebugServer	33
5.16.2.64 Width	33
5.17 Skylight.Miscellaneous.SmileyIds.Context Class Reference	33
5.17.1 Detailed Description	33
5.17.2 Member Data Documentation	33
5.17.2.1 Bird	33
5.18 Skylight.Crown Class Reference	34
5.18.1 Event Documentation	34
5.18.1.1 CrownEvent	34
5.19 Skylight.FaceChange Class Reference	34
5.19.1 Event Documentation	34
5.19.1.1 FaceEvent	34
5.20 Skylight.Miscellaneous.SmileyIds.Free Class Reference	35
5.20.1 Detailed Description	35
5.20.2 Member Data Documentation	35
5.20.2.1 Smile	35
5.21 Skylight.GetWoot Class Reference	35
5.21.1 Event Documentation	35
5.21.1.1 MagicCoinEvent	35
5.22 Skylight.GiveWizard2 Class Reference	36
5.22.1 Event Documentation	36
5.22.1.1 RedWizardEvent	36
5.23 Skylight.GodMode Class Reference	36
5.23.1 Event Documentation	36

5.23.1.1	GodEvent	36
5.24	Skylight.Grinch Class Reference	37
5.24.1	Event Documentation	37
5.24.1.1	GrinchEvent	37
5.25	Skylight.Hide Class Reference	37
5.25.1	Event Documentation	37
5.25.1.1	HideEvent	37
5.26	Skylight.In Class Reference	38
5.26.1	Detailed Description	39
5.26.2	Member Function Documentation	39
5.26.2.1	BlockEvent	39
5.26.2.2	ChatEvent	39
5.26.2.3	OnSignBlockEvent	39
5.26.2.4	PlayerEvent	40
5.26.2.5	RoomEvent	41
5.26.3	Event Documentation	41
5.26.3.1	AddEvent	41
5.26.3.2	CoinBlockEvent	41
5.26.3.3	InitEvent	41
5.27	Skylight.Info Class Reference	41
5.27.1	Event Documentation	41
5.27.1.1	InfoEvent	42
5.28	Skylight.Physics.ItemId Class Reference	42
5.28.1	Detailed Description	42
5.28.2	Member Function Documentation	42
5.28.2.1	IsBackgroundRotateable	42
5.28.2.2	IsClimbable	42
5.28.2.3	IsDecorationRotateable	43
5.28.2.4	IsSolid	43
5.29	Skylight.LeftWorld Class Reference	43
5.29.1	Event Documentation	44
5.29.1.1	LeaveEvent	44
5.30	Skylight.LevelChange Class Reference	44
5.30.1	Event Documentation	44
5.30.1.1	LevelUpEvent	44
5.31	Skylight.Meta Class Reference	44
5.31.1	Event Documentation	45
5.31.1.1	UpdateMetaEvent	45
5.32	Skylight.Moderator Class Reference	45
5.32.1	Event Documentation	45

5.32.1.1	ModModeEvent	45
5.33	Skylight.Move Class Reference	45
5.33.1	Event Documentation	46
5.33.1.1	JumpEvent	46
5.33.1.2	MovementEvent	46
5.34	Skylight.NoteBlock Class Reference	46
5.34.1	Event Documentation	46
5.34.1.1	SoundBlockEvent	46
5.35	Skylight.OnCoinGet Class Reference	46
5.35.1	Event Documentation	47
5.35.1.1	CoinCollectedEvent	47
5.36	Skylight.OnKill Class Reference	47
5.37	Skylight.Out Class Reference	47
5.37.1	Detailed Description	49
5.37.2	Member Function Documentation	49
5.37.2.1	Build	49
5.37.2.2	Build	49
5.37.2.3	Build	49
5.37.2.4	Clear	49
5.37.2.5	HoldDown	50
5.37.2.6	HoldLeft	51
5.37.2.7	HoldRight	51
5.37.2.8	HoldUp	51
5.37.2.9	InputCode	51
5.37.2.10	Jump	51
5.37.2.11	Kick	51
5.37.2.12	Kick	52
5.37.2.13	Loadlevel	52
5.37.2.14	Move	52
5.37.2.15	Move	52
5.37.2.16	Release	52
5.37.2.17	Reset	52
5.37.2.18	Respawn	52
5.37.2.19	Respawn	53
5.37.2.20	RespawnAll	53
5.37.2.21	Save	53
5.37.2.22	Say	53
5.37.2.23	SetAllPotionBans	53
5.37.2.24	SetCode	53
5.37.2.25	SetEdit	53

5.37.2.26 SetEdit	54
5.37.2.27 SetGod	54
5.37.2.28 SetMute	54
5.37.2.29 SetMute	54
5.37.2.30 SetPotionBan	54
5.37.2.31 SetSmiley	54
5.37.2.32 SetTitle	55
5.37.2.33 SetVisibility	55
5.37.2.34 Teleport	55
5.37.2.35 Teleport	55
5.37.2.36 TeleportAll	55
5.37.3 Property Documentation	55
5.37.3.1 Bot	55
5.37.3.2 C	56
5.37.3.3 R	56
5.38 Skylight.Blocks.PercussionBlock Class Reference	56
5.38.1 Detailed Description	56
5.38.2 Constructor & Destructor Documentation	57
5.38.2.1 PercussionBlock	57
5.38.3 Member Data Documentation	58
5.38.3.1 Base1	58
5.38.4 Property Documentation	58
5.38.4.1 PercussionId	58
5.39 Skylight.Blocks.PianoBlock Class Reference	58
5.39.1 Detailed Description	59
5.39.2 Constructor & Destructor Documentation	59
5.39.2.1 PianoBlock	59
5.39.3 Member Data Documentation	59
5.39.3.1 C1	59
5.39.4 Property Documentation	59
5.39.4.1 Pianold	59
5.40 Skylight.Player Class Reference	59
5.40.1 Detailed Description	62
5.40.2 Constructor & Destructor Documentation	62
5.40.2.1 Player	62
5.40.3 Member Function Documentation	62
5.40.3.1 Tick	62
5.40.4 Member Data Documentation	63
5.40.4.1 Admins	63
5.40.4.2 Boost	63

5.40.4.3	CheckpointX	63
5.40.4.4	CheckpointY	63
5.40.4.5	CurrentBlockId	63
5.40.4.6	Gravity	63
5.40.4.7	Isclubmember	63
5.40.4.8	ModifierX	63
5.40.4.9	ModifierY	63
5.40.4.10	MudBuoyancy	63
5.40.4.11	MudDrag	64
5.40.4.12	NoModifierDragX	64
5.40.4.13	NoModifierDragY	64
5.40.4.14	Size	64
5.40.4.15	SpeedX	64
5.40.4.16	SpeedY	64
5.40.4.17	SwitchOpened	64
5.40.4.18	WaterBuoyancy	64
5.40.4.19	WaterDrag	64
5.40.4.20	X	64
5.40.4.21	Y	64
5.40.5	Property Documentation	64
5.40.5.1	BlockX	64
5.40.5.2	BlockY	65
5.40.5.3	BlueCoins	65
5.40.5.4	Coins	65
5.40.5.5	CollectedMagic	65
5.40.5.6	DeathCount	65
5.40.5.7	HasAccess	65
5.40.5.8	HasBoost	65
5.40.5.9	HasChat	65
5.40.5.10	HasClub	65
5.40.5.11	HasCommandAccess	66
5.40.5.12	HasCrown	66
5.40.5.13	HasGravityModifier	66
5.40.5.14	HasSilverCrown	66
5.40.5.15	Id	66
5.40.5.16	IsBot	66
5.40.5.17	IsFriend	66
5.40.5.18	IsGod	66
5.40.5.19	IsHoldingDown	66
5.40.5.20	IsHoldingLeft	67

5.40.5.21 IsHoldingRight	67
5.40.5.22 IsHoldingSpace	67
5.40.5.23 IsHoldingUp	67
5.40.5.24 IsMod	67
5.40.5.25 IsOwner	67
5.40.5.26 Level	67
5.40.5.27 Name	67
5.40.5.28 PlayingIn	67
5.40.5.29 PotionEffects	68
5.40.5.30 Smiley	68
5.40.5.31 XpLevel	68
5.41 Skylight.Arguments.PlayerEventArgs Class Reference	68
5.41.1 Detailed Description	68
5.41.2 Constructor & Destructor Documentation	69
5.41.2.1 PlayerEventArgs	69
5.41.3 Property Documentation	69
5.41.3.1 Origin	69
5.41.3.2 RawMessage	69
5.41.3.3 Subject	69
5.42 Skylight.Blocks.PortalBlock Class Reference	69
5.42.1 Detailed Description	70
5.42.2 Constructor & Destructor Documentation	70
5.42.2.1 PortalBlock	70
5.42.3 Property Documentation	70
5.42.3.1 Direction	70
5.42.3.2 PortalDestination	70
5.42.3.3 PortalId	70
5.42.3.4 Visible	70
5.43 Skylight.Potion Class Reference	71
5.43.1 Event Documentation	71
5.43.1.1 PotionEvent	71
5.44 Skylight.Potions Class Reference	71
5.44.1 Event Documentation	71
5.44.1.1 PotionToggleEvent	71
5.45 Skylight.RefreshShop Class Reference	72
5.45.1 Event Documentation	72
5.45.1.1 RefreshshopEvent	72
5.46 Skylight.ResetWorld Class Reference	72
5.47 Skylight.Room Class Reference	72
5.47.1 Detailed Description	74

5.47.2	Constructor & Destructor Documentation	74
5.47.2.1	Room	74
5.47.3	Property Documentation	74
5.47.3.1	BlocksLoaded	74
5.47.3.2	BlueActivated	74
5.47.3.3	ChatLog	74
5.47.3.4	EditKey	74
5.47.3.5	GravityMultiplier	74
5.47.3.6	GreenActivated	74
5.47.3.7	HasPull	74
5.47.3.8	Height	75
5.47.3.9	Id	75
5.47.3.10	IsInitialized	75
5.47.3.11	IsTutorialRoom	75
5.47.3.12	JoinedRooms	75
5.47.3.13	Map	75
5.47.3.14	Name	75
5.47.3.15	OnlineBots	75
5.47.3.16	OnlinePlayers	75
5.47.3.17	Owner	76
5.47.3.18	Plays	76
5.47.3.19	PotionsAllowed	76
5.47.3.20	Pull	76
5.47.3.21	Pulls	76
5.47.3.22	Receiver	76
5.47.3.23	RedActivated	76
5.47.3.24	RoomKey	76
5.47.3.25	TimeDoorsVisible	76
5.47.3.26	TotalWoots	77
5.47.3.27	Width	77
5.47.3.28	Woots	77
5.48	Skylight.Arguments.RoomEventArgs Class Reference	77
5.48.1	Detailed Description	77
5.48.2	Constructor & Destructor Documentation	77
5.48.2.1	RoomEventArgs	77
5.48.3	Property Documentation	78
5.48.3.1	ChangedRoom	78
5.49	Skylight.Blocks.RoomPortalBlock Class Reference	78
5.49.1	Detailed Description	78
5.49.2	Constructor & Destructor Documentation	78

5.49.2.1	RoomPortalBlock	78
5.49.3	Property Documentation	79
5.49.3.1	PortalDestination	79
5.50	Skylight.Save Class Reference	79
5.50.1	Event Documentation	79
5.50.1.1	SavedEvent	79
5.51	Skylight.Miscellaneous.SmileyIds.Shop Class Reference	79
5.51.1	Detailed Description	80
5.51.2	Member Data Documentation	80
5.51.2.1	Ninja	80
5.52	Skylight.Show Class Reference	80
5.52.1	Event Documentation	80
5.52.1.1	ShowEvent	80
5.53	Skylight.SignBlock Class Reference	80
5.53.1	Member Function Documentation	81
5.53.1.1	OnSignBlockEvent	81
5.53.2	Event Documentation	81
5.53.2.1	SignBlockEvent	81
5.54	Skylight.Miscellaneous.SmileyIds Class Reference	81
5.54.1	Detailed Description	81
5.55	Skylight.Miscellaneous.SmileyIds.Special Class Reference	81
5.55.1	Detailed Description	82
5.55.2	Member Data Documentation	82
5.55.2.1	Superman	82
5.56	Skylight.Tele Class Reference	82
5.56.1	Event Documentation	82
5.56.1.1	DeathEvent	82
5.56.1.2	ResetEvent	82
5.57	Skylight.Teleport Class Reference	82
5.57.1	Event Documentation	83
5.57.1.1	TeleportEvent	83
5.58	Skylight.Blocks.TextBlock Class Reference	83
5.58.1	Detailed Description	83
5.58.2	Constructor & Destructor Documentation	84
5.58.2.1	TextBlock	84
5.58.3	Property Documentation	85
5.58.3.1	Text	85
5.59	Skylight.Trophy Class Reference	85
5.59.1	Event Documentation	85
5.59.1.1	TrophyEvent	85

5.60 Skylight.Miscellaneous.SmileYds.Unimplemented Class Reference	85
5.60.1 Detailed Description	86
5.60.2 Member Data Documentation	86
5.60.2.1 Dj	86
5.61 Skylight.Upgrade Class Reference	86
5.61.1 Event Documentation	86
5.61.1.1 UpdateEvent	86
5.62 Skylight.Witch Class Reference	86
5.62.1 Event Documentation	87
5.62.1.1 WitchEvent	87
5.63 Skylight.Wizard Class Reference	87
5.63.1 Event Documentation	87
5.63.1.1 WizardEvent	87
5.64 Skylight.WootUp Class Reference	87
5.64.1 Event Documentation	88
5.64.1.1 WootEvent	88
5.65 Skylight.Wp Class Reference	88
5.65.1 Event Documentation	88
5.65.1.1 RoomPortalBlockEvent	88
5.66 Skylight.Write Class Reference	88
5.66.1 Event Documentation	89
5.66.1.1 SystemMessageEvent	89
Index	90

Chapter 1

Namespace Index

1.1 Packages

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

Skylight	7
Skylight.Arguments	8
Skylight.Blocks	8
Skylight.Miscellaneous	9
Skylight.Physics	9
Skylight.Properties	9

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Skylight.Access	11
Skylight.Add	11
Skylight.AddSpecialBlock	12
Skylight.Autotext	12
Skylight.Miscellaneous.SmileyIds.Beta	13
Skylight.Blocks.Block	13
Skylight.Blocks.CoinBlock	23
Skylight.Blocks.PercussionBlock	56
Skylight.Blocks.PianoBlock	58
Skylight.Blocks.PortalBlock	69
Skylight.Blocks.RoomPortalBlock	78
Skylight.Blocks.TextBlock	83
Skylight.BlockChanged	15
Skylight.Chat	21
Skylight.ChatOld	22
Skylight.ClearMap	23
Skylight.CoinObject	24
Skylight.Miscellaneous.SmileyIds.Contest	33
Skylight.Crown	34
EventArgs	
Skylight.Arguments.BlockEventArgs	15
Skylight.Arguments.ChatEventArgs	21
Skylight.Arguments.PlayerEventArgs	68
Skylight.Arguments.RoomEventArgs	77
Skylight.FaceChange	34
Skylight.Miscellaneous.SmileyIds.Free	35
Skylight.GetWoot	35
Skylight.GiveWizard2	36
Skylight.GodMode	36
Skylight.Grinch	37
Skylight.Hide	37
Skylight.In	38
Skylight.Info	41
Skylight.LeftWorld	43
Skylight.LevelChange	44
Skylight.Meta	44
Skylight.Moderator	45
Skylight.Move	45

Skylight.NoteBlock	46
object	
Skylight.Physics.Config	25
Skylight.Physics.ItemId	42
Skylight.OnCoinGet	46
Skylight.OnKill	47
Skylight.Out	47
Skylight.Player	59
Skylight.Bot	17
Skylight.Potion	71
Skylight.Potions	71
Skylight.RefreshShop	72
Skylight.ResetWorld	72
Skylight.Room	72
Skylight.Save	79
Skylight.Miscellaneous.SmileyIds.Shop	79
Skylight.Show	80
Skylight.SignBlock	80
Skylight.Miscellaneous.SmileyIds	81
Skylight.Miscellaneous.SmileyIds.Special	81
Skylight.Tele	82
Skylight.Teleport	82
Skylight.Trophy	85
Skylight.Miscellaneous.SmileyIds.Unimplemented	85
Skylight.Upgrade	86
Skylight.Witch	86
Skylight.Wizard	87
Skylight.WootUp	87
Skylight.Wp	88
Skylight.Write	88

Chapter 3

Class Index

3.1 Class List

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

Skylight.Access	11
Skylight.Add	11
Skylight.AddSpecialBlock	12
Skylight.Autotext	12
Skylight.Miscellaneous.SmileyIds.Beta	
Class Beta .	13
Skylight.Blocks.Block	
Class Block .	13
Skylight.BlockChanged	15
Skylight.Arguments.BlockEventArgs	
Sends the delegate an event based on the content of the block.	15
Skylight.Bot	
The entire bot (main core). This is what the user instantiates when they want a new bot.	17
Skylight.Chat	21
Skylight.Arguments.ChatEventArgs	
The class that handles all chat-based messages from the server including ones sent to the user through system.	21
Skylight.ChatOld	22
Skylight.ClearMap	23
Skylight.Blocks.CoinBlock	
Class CoinBlock .	23
Skylight.CoinObject	24
Skylight.Physics.Config	
Class Configuration .	25
Skylight.Miscellaneous.SmileyIds.Contest	
Class Contest .	33
Skylight.Crown	34
Skylight.FaceChange	34
Skylight.Miscellaneous.SmileyIds.Free	
Class Free .	35
Skylight.GetWoot	35
Skylight.GiveWizard2	36
Skylight.GodMode	36
Skylight.Grinch	37
Skylight.Hide	37
Skylight.In	
The main class that takes in events from the playerio client.	38
Skylight.Info	41

Skylight.Physics.ItemId	
Class item id.	42
Skylight.LeftWorld	43
Skylight.LevelChange	44
Skylight.Meta	44
Skylight.Moderator	45
Skylight.Move	45
Skylight.NoteBlock	46
Skylight.OnCoinGet	46
Skylight.OnKill	47
Skylight.Out	
Class Out. This class sends information to the server.	47
Skylight.Blocks.PercussionBlock	
Class PercussionBlock.	56
Skylight.Blocks.PianoBlock	
Class PianoBlock.	58
Skylight.Player	
Class Player.	59
Skylight.Arguments.PlayerEventArgs	
This class sets the properties about a player such as who the player is (username), their most recent message and where they are (the room that they are in).	68
Skylight.Blocks.PortalBlock	
Class PortalBlock.	69
Skylight.Potion	71
Skylight.Potions	71
Skylight.RefreshShop	72
Skylight.ResetWorld	72
Skylight.Room	
Class Room.	72
Skylight.Arguments.RoomEventArgs	
Class RoomEventArgs.	77
Skylight.Blocks.RoomPortalBlock	
Class RoomPortalBlock.	78
Skylight.Save	79
Skylight.Miscellaneous.SmileyIds.Shop	
Class Shop.	79
Skylight.Show	80
Skylight.SignBlock	80
Skylight.Miscellaneous.SmileyIds	
Class SmileyIds.	81
Skylight.Miscellaneous.SmileyIds.Special	
Class Special.	81
Skylight.Tele	82
Skylight.Teleport	82
Skylight.Blocks.TextBlock	
Class TextBlock.	83
Skylight.Trophy	85
Skylight.Miscellaneous.SmileyIds.Unimplemented	
Class Unimplemented.	85
Skylight.Upgrade	86
Skylight.Witch	86
Skylight.Wizard	87
Skylight.WootUp	87
Skylight.Wp	88
Skylight.Write	88

Chapter 4

Namespace Documentation

4.1 Package Skylight

Namespaces

- package [Arguments](#)
- package [Blocks](#)
- package [Miscellaneous](#)
- package [Physics](#)
- package [Properties](#)

Classes

- class [Access](#)
- class [Add](#)
- class [AddSpecialBlock](#)
- class [Autotext](#)
- class [BlockChanged](#)
- class [Bot](#)

The entire bot (main core). This is what the user instantiates when they want a new bot.

- class [Chat](#)
- class [ChatOld](#)
- class [ClearMap](#)
- class [CoinObject](#)
- class [Crown](#)
- class [FaceChange](#)
- class [GetWoot](#)
- class [GiveWizard2](#)
- class [GodMode](#)
- class [Grinch](#)
- class [Hide](#)
- class [In](#)

The main class that takes in events from the playerio client.

- class [Info](#)
- class [LeftWorld](#)
- class [LevelChange](#)
- class [Meta](#)
- class [Moderator](#)
- class [Move](#)

- class [NoteBlock](#)
- class [OnCoinGet](#)
- class [OnKill](#)
- class [Out](#)

Class [Out](#). This class sends information to the server.

- class [Player](#)

Class [Player](#).

- class [Potion](#)
- class [Potions](#)
- class [RefreshShop](#)
- class [ResetWorld](#)
- class [Room](#)

Class [Room](#).

- class [Save](#)
- class [Show](#)
- class [SignBlock](#)
- class [Tele](#)
- class [Teleport](#)
- class [Trophy](#)
- class [Upgrade](#)
- class [Witch](#)
- class [Wizard](#)
- class [WootUp](#)
- class [Wp](#)
- class [Write](#)

4.2 Package Skylight.Arguments

Classes

- class [BlockEventArgs](#)

Sends the delegate an event based on the content of the block.

- class [ChatEventArgs](#)

The class that handles all chat-based messages from the server including ones sent to the user through system.

- class [PlayerEventArgs](#)

This class sets the properties about a player such as who the player is (username), their most recent message and where they are (the room that they are in).

- class [RoomEventArgs](#)

Class [RoomEventArgs](#).

4.3 Package Skylight.Blocks

Classes

- class [Block](#)

Class [Block](#).

- class [BlockIds](#)

Class [BlockIds](#).

- class [CoinBlock](#)

Class [CoinBlock](#).

- class [PercussionBlock](#)

- Class [PercussionBlock](#).*
- class [PianoBlock](#)
Class [PianoBlock](#).
- class [PortalBlock](#)
Class [PortalBlock](#).
- class [RoomPortalBlock](#)
Class [RoomPortalBlock](#).
- class [TextBlock](#)
Class [TextBlock](#).

4.4 Package Skylight.Miscellaneous

Classes

- class **PotionIds**
Class [PotionIds](#).
- class [SmileyIds](#)
Class [SmileyIds](#).
- class **Tools**
Tools that are available to the core of the program (converting a player id or name into a player object) and internal methods are mostly stored here.

4.5 Package Skylight.Physics

Classes

- class [Config](#)
Class [Configuration](#).
- class [ItemId](#)
Class [item id](#).

4.6 Package Skylight.Properties

Classes

- class **Settings**

Chapter 5

Class Documentation

5.1 Skylight.Access Class Reference

Public Member Functions

- **Access** ([In](#) @in)
- void **OnAccess** (Message m)

Events

- [In.PlayerEvent](#) [GainAccessEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.1.1 Event Documentation

5.1.1.1 [In.PlayerEvent](#) Skylight.Access.GainAccessEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Access.cs

5.2 Skylight.Add Class Reference

Public Member Functions

- **Add** ([In](#) @in)
- void **OnAdd** (Message m)

Events

- [In.PlayerEvent](#) **OnAddEvent** = delegate { }

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

- Skylight/Add.cs

5.3 Skylight.AddSpecialBlock Class Reference

Public Member Functions

- **AddSpecialBlock** ([In](#) @in)
- void **OnAddScifiOrSpikes** (Message m)

Events

- [In.BlockEvent](#) [RotateEvent](#) = delegate { }
- All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).*

5.3.1 Event Documentation

5.3.1.1 [In.BlockEvent](#) Skylight.AddSpecialBlock.RotateEvent = delegate { }

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

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

- Skylight/AddSpecialBlock.cs

5.4 Skylight.Autotext Class Reference

Public Member Functions

- **Autotext** ([In](#) @in)
- void **OnAutotext** (Message m)

Events

- [In.ChatEvent](#) [AutotextEvent](#) = delegate { }
- All of the delegates for ChatEvent. [Chat](#) events are when the player says something, and distinguishes between auto text and system messages and much more.*

5.4.1 Event Documentation

5.4.1.1 [In.ChatEvent](#) Skylight.Autotext.AutotextEvent = delegate { }

All of the delegates for ChatEvent. [Chat](#) events are when the player says something, and distinguishes between auto text and system messages and much more.

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

- Skylight/Autotext.cs

5.5 Skylight.Miscellaneous.SmileyIds.Beta Class Reference

Class [Beta](#).

Public Attributes

- const int [Tear](#) = 6
The types of beta smilies.

5.5.1 Detailed Description

Class [Beta](#).

5.5.2 Member Data Documentation

5.5.2.1 const int Skylight.Miscellaneous.SmileyIds.Beta.Tear = 6

The types of beta smilies.

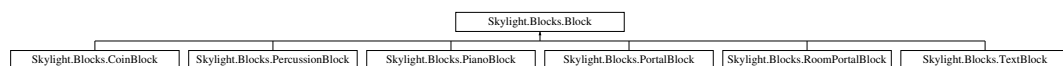
The documentation for this class was generated from the following file:

- Skylight/Miscellaneous/SmileyIds.cs

5.6 Skylight.Blocks.Block Class Reference

Class [Block](#).

Inheritance diagram for Skylight.Blocks.Block:



Public Member Functions

- [Block](#) (int id, int x, int y, int z=0, int direction=Up)
Initializes a new instance of the [Block](#) class.

Public Attributes

- const int [Right](#) = 1
The directions of the block.

Properties

- bool [IsSolid](#) [get]
Gets a value indicating whether this instance is solid.
- int [Direction](#) [get, set]
Gets or sets the direction.
- int [Id](#) [get, set]

- Gets the identifier.*
- `int Z` [get, set]
Gets the z coordinate (background or foreground)
- `int X` [get, set]
Gets the x coordinate
- `int Y` [get, set]
Gets the y coordinate
- `Player Placer` [get, set]
Gets the placer (the [Player](#) who placed the block).

5.6.1 Detailed Description

Class [Block](#).

5.6.2 Constructor & Destructor Documentation

5.6.2.1 `Skylight.Blocks.Block.Block (int id, int x, int y, int z = 0, int direction = Up)`

Initializes a new instance of the [Block](#) class.

Parameters

<i>id</i>	The id of the block.
<i>x</i>	The x coord.
<i>y</i>	The y coord.
<i>z</i>	The z coord.
<i>direction</i>	The direction.

5.6.3 Member Data Documentation

5.6.3.1 `const int Skylight.Blocks.Block.Right = 1`

The directions of the block.

5.6.4 Property Documentation

5.6.4.1 `int Skylight.Blocks.Block.Direction` [get], [set]

Gets or sets the direction.

The direction.

5.6.4.2 `int Skylight.Blocks.Block.Id` [get], [set]

Gets the identifier.

The identifier.

5.6.4.3 `bool Skylight.Blocks.Block.IsSolid` [get]

Gets a value indicating whether this instance is solid.

`true` if this instance is solid; otherwise, `false`.

5.6.4.4 Player Skylight.Blocks.Block.Placer [get], [set]

Gets the placer (the [Player](#) who placed the block).

The placer.

5.6.4.5 int Skylight.Blocks.Block.X [get], [set]

Gets the x coordinate

The x.

5.6.4.6 int Skylight.Blocks.Block.Y [get], [set]

Gets the y coordinate

The y.

5.6.4.7 int Skylight.Blocks.Block.Z [get], [set]

Gets the z coordinate (background or foreground)

The z.

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

- Skylight/Blocks/Block.cs

5.7 Skylight.BlockChanged Class Reference

Public Member Functions

- **BlockChanged** ([In](#) @in)
- void **OnBlock** (Message m)

Events

- [In.BlockEvent](#) [NormalBlockEvent](#) = delegate { }

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

5.7.1 Event Documentation

5.7.1.1 In.BlockEvent Skylight.BlockChanged.NormalBlockEvent = delegate { }

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

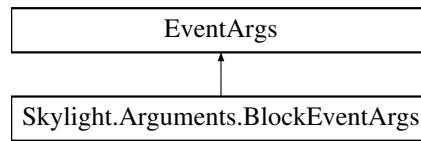
The documentation for this class was generated from the following file:

- Skylight/BlockChanged.cs

5.8 Skylight.Arguments.BlockEventArgs Class Reference

Sends the delegate an event based on the content of the block.

Inheritance diagram for Skylight.Arguments.BlockEventArgs:



Public Member Functions

- **BlockEventArgs** (**Block** b, **Room** origin=null)
*Initializes a new instance of the **BlockEventArgs**. Send a block changed event.*

Properties

- **Room Origin** [get, set]
The room object (with room id).
- **Block Placed** [get]
A placed block.
- **Player Placer** [get]
The player who placed the block (see Placed).

5.8.1 Detailed Description

Sends the delegate an event based on the content of the block.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Skylight.Arguments.BlockEventArgs.BlockEventArgs (**Block** b, **Room** origin = null)

Initializes a new instance of the **BlockEventArgs**. Send a block changed event.

Parameters

<i>b</i>	The block
<i>origin</i>	The room where the block originated from.

5.8.3 Property Documentation

5.8.3.1 Room Skylight.Arguments.BlockEventArgs.Origin [get], [set]

The room object (with room id).

5.8.3.2 Block Skylight.Arguments.BlockEventArgs.Placed [get]

A placed block.

5.8.3.3 Player Skylight.Arguments.BlockEventArgs.Placer [get]

The player who placed the block (see Placed).

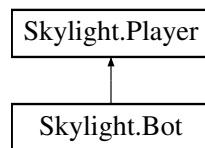
The documentation for this class was generated from the following file:

- Skylight/Arguments/BlockEventArgs.cs

5.9 Skylight.Bot Class Reference

The entire bot (main core). This is what the user instantiates when they want a new bot.

Inheritance diagram for Skylight.Bot:



Public Types

- enum [AccountType](#) : sbyte { [AccountType.Regular](#) = 0, [AccountType.Facebook](#) = 1, [AccountType.Kongregate](#) = 2, [AccountType.ArmorGames](#) = 3 }

All of the possible account types. Defaults to Regular if unknown.

Public Member Functions

- [Bot](#) ([Room](#) r, string emailOrToken=Tools.GuestEmail, string passwordOrToken=Tools.GuestPassword, [AccountType](#) accType=[AccountType.Regular](#))

The main bot class.

- void [Login](#) ()

The main method to login the bot with the credentials already specified.

- void [Join](#) (bool createRoom=true)

Join the room that was already set.

- void [Disconnect](#) ()

Disconnect the bot.

Properties

- bool [IsConnected](#) [get, set]

Whether or not the bot is connected to the world.

- bool [Joined](#) [get, set]

The room the bot is joined to.

- bool [ShouldTick](#) [get, set]

If the physics clock should tick or not.

- Client [Client](#) [get, set]

The PlayerIO client class.

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

The delay between block messages to the server in milliseconds.

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

- The delay between speech messages to the server in milliseconds.*
- string [ChatPrefix](#) [get, set]
The prefix to add to all outgoing chat messages.
- [Out Push](#) [get, set]
The object where the events go to the server.
- [Room R](#) [get, set]
The current room object.
- Connection [Connection](#) [get, set]
The active connection object to the room.
- static [Room CurrentRoom](#) [get, set]
The current room that the bot is in.

Additional Inherited Members

5.9.1 Detailed Description

The entire bot (main core). This is what the user instantiates when they want a new bot.

5.9.2 Member Enumeration Documentation

5.9.2.1 enum [Skylight.Bot.AccountType](#) : sbyte

All of the possible account types. Defaults to Regular if unknown.

Enumerator

- Regular** The regular login method via email and password on the official everybodyedits.com website.
- Facebook** Facebook login using an auth token.
- Kongregate** Kongregate login using a kongregate user id (an integer) and an auth token.
- ArmorGames** ArmorGames login using a user token and a password token, both of which are 32 chars long and hex.

5.9.3 Constructor & Destructor Documentation

5.9.3.1 [Skylight.Bot.Bot](#) ([Room](#) *r*, string *emailOrToken* = [Tools.GuestEmail](#), string *passwordOrToken* = [Tools.GuestPassword](#), [AccountType](#) *accType* = [AccountType.Regular](#))

The main bot class.

Parameters

<i>r</i>	The room object that the bot will join.
<i>emailOrToken</i>	The email or token.
<i>passwordOrToken</i>	The password or token.
<i>accType</i>	The account type. Default is regular but it automatically guesses.

5.9.4 Member Function Documentation

5.9.4.1 void [Skylight.Bot.Disconnect](#) ()

Disconnect the bot.

5.9.4.2 void Skylight.Bot.Join (bool *createRoom* = true)

Join the room that was already set.

Parameters

<i>createRoom</i>	
-------------------	--

5.9.4.3 void Skylight.Bot.Login ()

The main method to login the bot with the credentials already specified.

5.9.5 Property Documentation

5.9.5.1 int Skylight.Bot.BlockDelay [get], [set]

The delay between block messages to the server in milliseconds.

5.9.5.2 string Skylight.Bot.ChatPrefix [get], [set]

The prefix to add to all outgoing chat messages.

5.9.5.3 Client Skylight.Bot.Client [get], [set]

The PlayerIO client class.

5.9.5.4 Connection Skylight.Bot.Connection [get], [set]

The active connection object to the room.

5.9.5.5 Room Skylight.Bot.CurrentRoom [static], [get], [set]

The current room that the bot is in.

5.9.5.6 bool Skylight.Bot.IsConnected [get], [set]

Whether or not the bot is connected to the world.

5.9.5.7 bool Skylight.Bot.Joined [get], [set]

The room the bot is joined to.

5.9.5.8 Out Skylight.Bot.Push [get], [set]

The object where the events go to the server.

5.9.5.9 Room Skylight.Bot.R [get], [set]

The current room object.

5.9.5.10 bool Skylight.Bot.ShouldTick [get], [set]

If the physics clock should tick or not.

5.9.5.11 `int Skylight.Bot.SpeechDelay` `[get]`, `[set]`

The delay between speech messages to the server in milliseconds.

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

- Skylight/Bot.cs

5.10 Skylight.Chat Class Reference

Public Member Functions

- **Chat** (`In @in`)
- void **OnSay** (`Message m`)

Events

- `In.ChatEvent NormalChatEvent` = `delegate { }`

All of the delegates for ChatEvent. [Chat](#) events are when the player says something, and distinguishes between auto text and system messages and much more.

5.10.1 Event Documentation

5.10.1.1 `In.ChatEvent Skylight.Chat.NormalChatEvent` = `delegate { }`

All of the delegates for ChatEvent. [Chat](#) events are when the player says something, and distinguishes between auto text and system messages and much more.

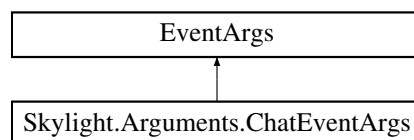
The documentation for this class was generated from the following file:

- Skylight/Chat.cs

5.11 Skylight.Arguments.ChatEventArgs Class Reference

The class that handles all chat-based messages from the server including ones sent to the user through system.

Inheritance diagram for Skylight.Arguments.ChatEventArgs:



Public Member Functions

- `ChatEventArgs` (`Player` speaker, `Room` origin)

The main method where the chat messages are sent. This method sets the properties in this class to the speaker and origin of the message, where it is handed off to a delegate later.

Properties

- string [Message](#) [get]
The message object containing the message content.
- [Room Origin](#) [get]
The origin (room) where the message came from.
- [Player Speaker](#) [get]
Who said the message (player).

5.11.1 Detailed Description

The class that handles all chat-based messages from the server including ones sent to the user through system.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 Skylight.Arguments.ChatEventArgs.ChatEventArgs (*Player speaker*, *Room origin*)

The main method where the chat messages are sent. This method sets the properties in this class to the speaker and origin of the message, where it is handed off to a delegate later.

Parameters

<i>speaker</i>	The player who said the message.
<i>origin</i>	The room where the message originated.

5.11.3 Property Documentation

5.11.3.1 string Skylight.Arguments.ChatEventArgs.Message [get]

The message object containing the message content.

5.11.3.2 Room Skylight.Arguments.ChatEventArgs.Origin [get]

The origin (room) where the message came from.

5.11.3.3 Player Skylight.Arguments.ChatEventArgs.Speaker [get]

Who said the message (player).

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

- Skylight/Arguments/ChatEventArgs.cs

5.12 Skylight.ChatOld Class Reference

Public Member Functions

- **ChatOld** (In @in)
- void **OnSayOld** (Message m)

Events

- [In.ChatEvent SayOldEvent](#) = delegate { }

All of the delegates for ChatEvent. [Chat](#) events are when the player says something, and distinguishes between auto text and system messages and much more.

5.12.1 Event Documentation

5.12.1.1 In.ChatEvent Skylight.ChatOld.SayOldEvent = delegate { }

All of the delegates for ChatEvent. [Chat](#) events are when the player says something, and distinguishes between auto text and system messages and much more.

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

- Skylight/ChatOld.cs

5.13 Skylight.ClearMap Class Reference

Public Member Functions

- **ClearMap** ([In @in](#))
- void **OnClear** ()

Events

- [In.RoomEvent ClearEvent](#) = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.13.1 Event Documentation

5.13.1.1 In.RoomEvent Skylight.ClearMap.ClearEvent = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

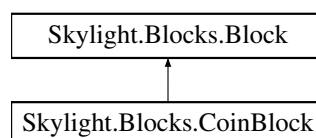
The documentation for this class was generated from the following file:

- Skylight/ClearMap.cs

5.14 Skylight.Blocks.CoinBlock Class Reference

Class [CoinBlock](#).

Inheritance diagram for Skylight.Blocks.CoinBlock:



Public Member Functions

- [CoinBlock](#) (int x, int y, int coinsRequired, bool isGate)
Initializes a new instance of the [CoinBlock](#) class.

Properties

- bool [IsGate](#) [get, set]
Gets a value indicating whether this instance is a gate.
- int [CoinsRequired](#) [get, set]
Gets the coins required to open the gate (or close the door).

Additional Inherited Members

5.14.1 Detailed Description

Class [CoinBlock](#).

5.14.2 Constructor & Destructor Documentation

5.14.2.1 Skylight.Blocks.CoinBlock.CoinBlock (int x, int y, int *coinsRequired*, bool *isGate*)

Initializes a new instance of the [CoinBlock](#) class.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.
<i>coinsRequired</i>	The coins required to open the gate if applicable.
<i>isGate</i>	if set to <code>true</code> [is gate].

5.14.3 Property Documentation

5.14.3.1 int Skylight.Blocks.CoinBlock.CoinsRequired [get], [set]

Gets the coins required to open the gate (or close the door).

The coins required.

5.14.3.2 bool Skylight.Blocks.CoinBlock.IsGate [get], [set]

Gets a value indicating whether this instance is a gate.

`true` if this instance is gate; otherwise, `false`.

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

- Skylight/Blocks/CoinBlock.cs

5.15 Skylight.CoinObject Class Reference

Public Member Functions

- [CoinObject](#) (In @in)

- void **OnAddCoinDoorOrGate** (Message m)

Events

- [In.BlockEvent](#) **CoinBlockEvent** = delegate { }

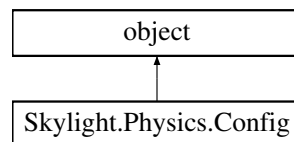
The documentation for this class was generated from the following file:

- Skylight/CoinObject.cs

5.16 Skylight.Physics.Config Class Reference

Class Configuration.

Inheritance diagram for Skylight.Physics.Config:



Static Public Attributes

- static string [PlayerioGameld](#) = "everybody-edits-su9rn58o40itdbnw69plyw"
The PlayerIO game id.
- static int [ServerTypeVersion](#) = 176
The server version.
- static string [ServerTypeNormalroom](#) = "Everybodyedits" + ServerTypeVersion
The server_type_normalroom
- static string [ServerTypeBetaroom](#) = "Beta" + ServerTypeVersion
The beta room
- static string [ServerTypeGuestserviceroom](#) = "LobbyGuest" + ServerTypeVersion
The guest only room
- static string [ServerTypeServiceroom](#) = "Lobby" + ServerTypeVersion
The service room
- static string [ServerTypeAuthroom](#) = "Auth" + ServerTypeVersion
The authentication room (temp)
- static string [ServerTypeBlacklistroom](#) = "QuickInviteHandler" + ServerTypeVersion
The blacklisted rooms
- static string [ServerTypeTutorialroom](#) = "Tutorial" + ServerTypeVersion + "_world_"
The tutorial room
- static string [ServerTypeTrackingroom](#) = "Tracking" + ServerTypeVersion
The tracking room
- static string [UrlBlog](#) = "http://blog.everybodyedits.com"
The url for the Everybody Edits blog
- static string [UrlClubmemberAboutPage](#) = "http://everybodyedits.com/club"
The url to the club member page
- static string [UrlTermsPage](#) = "http://everybodyedits.com/terms"
The terms of service website

- static string `UrlHelpPage` = "http://everybodyedits.com/help"
The help website
- static bool `UseDebugServer` = false
Whether or not to use the debug server
- static bool `RunInDevelopmentMode` = false
Whether or not to run in development mode
- static bool `ShowDisabledShopitems` = false
The show_disabled_shopitems
- static string `DevelopmentModeAutojoinRoom` = "PWvOaRlelvbUI"
The development_mode_autojoin_room
- static string `DebugNews` = ""
The debug_news
- static string `DeveloperServer` = "127.0.0.1:8184"
The developer_server
- static bool `ForceArmor` = false
The force armor authentication option
- static string `ArmorUserid` = null
The armor_userid
- static string `ArmorAuthtoken` = null
The armor_authtoken
- static bool `ForceMouseBreaker` = false
The force mouse breaker
- static string `MousebreakerAuthtoken` = null
The mousebreaker_authtoken
- static bool `ForceBeta` = false
The force beta
- static bool `ShowDebugProfile` = true
The show_debug_profile
- static string `DebugProfile` = ""
The debug_profile
- static bool `DisableCookie` = false
The disable cookie
- static bool `ShowDebugFriendrequest` = false
The show_debug_friendrequest
- static string `DebugFriendrequest` = ""
The debug_friendrequest
- static bool `ShowBlacklistInvitation` = false
The show_blacklist_invitation
- static string `DebugInvitation` = ""
The debug_invitation
- static int `PhysicsMsPerTick` = 10
The physics_ms_per_tick
- static double `PhysicsVariableMultiplier` = 7.752
The physics_variable_multiplier
- static double `PhysicsBaseDrag` = Math.Pow(0.9981, `PhysicsMsPerTick`)*1.00016
The physics_base_drag
- static double `PhysicsNoModifierDrag` = Math.Pow(0.99, `PhysicsMsPerTick`)*1.00016
The physics_no_modifier_drag
- static double `PhysicsWaterDrag` = Math.Pow(0.995, `PhysicsMsPerTick`)*1.00016
The physics_water_drag
- static double `PhysicsMudDrag` = Math.Pow(0.975, `PhysicsMsPerTick`)*1.00016

- The physics_mud_drag*
- static double [PhysicsJumpHeight](#) = 26
- The physics_jump_height*
- static double [PhysicsGravity](#) = 2
- The physics_gravity*
- static double [PhysicsBoost](#) = 16
- The physics_boost*
- static double [PhysicsWaterBuoyancy](#) = -0.5
- The physics_water_buoyancy*
- static double [PhysicsMudBuoyancy](#) = 0.4
- The physics_mud_buoyancy*
- static int [PhysicsQueueLength](#) = 2
- The physics_queue_length*
- static int [ShopPotionMax](#) = 10
- The shop_potion_max*
- static double [CameraLag](#) = 0.0625
- The camera_lag*
- static bool [IsMobile](#) = false
- The is mobile*
- static bool [EnableDebugShadow](#) = false
- The enable debug shadow*
- static int [Maxwidth](#) = 850
- The maxwidth*
- static int [Minwidth](#) = 640
- The minwidth*
- static int [Width](#) = 640
- The width*
- static int [Height](#) = 500
- The height*
- static int [MaxFrameRate](#) = 120
- The maximum frame rate*
- static int [MaxDailyWoot](#) = 10
- The max_daily_woot*
- static uint [GuestColor](#) = 3355443
- The guest_color*
- static uint [DefaultColor](#) = 15658734
- The default_color*
- static uint [DefaultColorDark](#) = 13421772
- The default_color_dark*
- static uint [FriendColor](#) = 65280
- The friend_color*
- static uint [FriendColorDark](#) = 47872
- The friend_color_dark*
- static uint [ModColor](#) = 16759552
- The mod_color*
- static uint [AdminColor](#) = 16757760
- The admin_color*
- static string[] [TutorialNames](#) = {"Moving", "Gravity", "Edit"}
- The tutorial_names*
- static bool [DisableTracking](#) = false
- The disable_tracking*

5.16.1 Detailed Description

Class Configuration.

5.16.2 Member Data Documentation

5.16.2.1 `uint Skylight.Physics.Config.AdminColor = 16757760` `[static]`

The `admin_color`

5.16.2.2 `string Skylight.Physics.Config.ArmorAuthtoken = null` `[static]`

The `armor_authtoken`

5.16.2.3 `string Skylight.Physics.Config.ArmorUserid = null` `[static]`

The `armor_userid`

5.16.2.4 `double Skylight.Physics.Config.CameraLag = 0.0625` `[static]`

The `camera_lag`

5.16.2.5 `string Skylight.Physics.Config.DebugFriendrequest = ""` `[static]`

The `debug_friendrequest`

5.16.2.6 `string Skylight.Physics.Config.DebugInvitation = ""` `[static]`

The `debug_invitation`

5.16.2.7 `string Skylight.Physics.Config.DebugNews = ""` `[static]`

The `debug_news`

5.16.2.8 `string Skylight.Physics.Config.DebugProfile = ""` `[static]`

The `debug_profile`

5.16.2.9 `uint Skylight.Physics.Config.DefaultColor = 15658734` `[static]`

The `default_color`

5.16.2.10 `uint Skylight.Physics.Config.DefaultColorDark = 13421772` `[static]`

The `default_color_dark`

5.16.2.11 `string Skylight.Physics.Config.DeveloperServer = "127.0.0.1:8184"` `[static]`

The `developer_server`

5.16.2.12 `string Skylight.Physics.Config.DevelopmentModeAutojoinRoom = "PWvOaRlelvbUI"` `[static]`

The development_mode_autojoin_room

5.16.2.13 `bool Skylight.Physics.Config.DisableCookie = false` `[static]`

The disable cookie

5.16.2.14 `bool Skylight.Physics.Config.DisableTracking = false` `[static]`

The disable_tracking

5.16.2.15 `bool Skylight.Physics.Config.EnableDebugShadow = false` `[static]`

The enable debug shadow

5.16.2.16 `bool Skylight.Physics.Config.ForceArmor = false` `[static]`

The force armor authentication option

5.16.2.17 `bool Skylight.Physics.Config.ForceBeta = false` `[static]`

The force beta

5.16.2.18 `bool Skylight.Physics.Config.ForceMouseBreaker = false` `[static]`

The force mouse breaker

5.16.2.19 `uint Skylight.Physics.Config.FriendColor = 65280` `[static]`

The friend_color

5.16.2.20 `uint Skylight.Physics.Config.FriendColorDark = 47872` `[static]`

The friend_color_dark

5.16.2.21 `uint Skylight.Physics.Config.GuestColor = 3355443` `[static]`

The guest_color

5.16.2.22 `int Skylight.Physics.Config.Height = 500` `[static]`

The height

5.16.2.23 `bool Skylight.Physics.Config.IsMobile = false` `[static]`

The is mobile

5.16.2.24 `int Skylight.Physics.Config.MaxDailyWoot = 10` `[static]`

The `max_daily_woot`

5.16.2.25 `int Skylight.Physics.Config.MaxFrameRate = 120` `[static]`

The maximum frame rate

5.16.2.26 `int Skylight.Physics.Config.Maxwidth = 850` `[static]`

The `maxwidth`

5.16.2.27 `int Skylight.Physics.Config.Minwidth = 640` `[static]`

The `minwidth`

5.16.2.28 `uint Skylight.Physics.Config.ModColor = 16759552` `[static]`

The `mod_color`

5.16.2.29 `string Skylight.Physics.Config.MousebreakerAuthtoken = null` `[static]`

The `mousebreaker_authtoken`

5.16.2.30 `double Skylight.Physics.Config.PhysicsBaseDrag = Math.Pow(0.9981, PhysicsMsPerTick)*1.00016` `[static]`

The `physics_base_drag`

5.16.2.31 `double Skylight.Physics.Config.PhysicsBoost = 16` `[static]`

The `physics_boost`

5.16.2.32 `double Skylight.Physics.Config.PhysicsGravity = 2` `[static]`

The `physics_gravity`

5.16.2.33 `double Skylight.Physics.Config.PhysicsJumpHeight = 26` `[static]`

The `physics_jump_height`

5.16.2.34 `int Skylight.Physics.Config.PhysicsMsPerTick = 10` `[static]`

The `physics_ms_per_tick`

5.16.2.35 `double Skylight.Physics.Config.PhysicsMudBuoyancy = 0.4` `[static]`

The `physics_mud_buoyancy`

5.16.2.36 `double Skylight.Physics.Config.PhysicsMudDrag = Math.Pow(0.975, PhysicsMsPerTick)*1.00016` [static]

The physics_mud_drag

5.16.2.37 `double Skylight.Physics.Config.PhysicsNoModifierDrag = Math.Pow(0.99, PhysicsMsPerTick)*1.00016`
[static]

The physics_no_modifier_drag

5.16.2.38 `int Skylight.Physics.Config.PhysicsQueueLength = 2` [static]

The physics_queue_length

5.16.2.39 `double Skylight.Physics.Config.PhysicsVariableMultiplier = 7.752` [static]

The physics_variable_multiplier

5.16.2.40 `double Skylight.Physics.Config.PhysicsWaterBuoyancy = -0.5` [static]

The physics_water_buoyancy

5.16.2.41 `double Skylight.Physics.Config.PhysicsWaterDrag = Math.Pow(0.995, PhysicsMsPerTick)*1.00016` [static]

The physics_water_drag

5.16.2.42 `string Skylight.Physics.Config.PlayerioGameId = "everybody-edits-su9rn58o40itdbnw69plyw"` [static]

The PlayerIO game id.

5.16.2.43 `bool Skylight.Physics.Config.RunInDevelopmentMode = false` [static]

Whether or not to run in development mode

5.16.2.44 `string Skylight.Physics.Config.ServerTypeAuthroom = "Auth" + ServerTypeVersion` [static]

The authentication room (temp)

5.16.2.45 `string Skylight.Physics.Config.ServerTypeBetaroom = "Beta" + ServerTypeVersion` [static]

The beta room

5.16.2.46 `string Skylight.Physics.Config.ServerTypeBlacklistroom = "QuickInviteHandler" + ServerTypeVersion` [static]

The blacklisted rooms

5.16.2.47 `string Skylight.Physics.Config.ServerTypeGuestserviceroom = "LobbyGuest" + ServerTypeVersion` [static]

The guest only room

5.16.2.48 `string Skylight.Physics.Config.ServerTypeNormalroom = "Everybodyedits" + ServerTypeVersion` `[static]`

The server_type_normalroom

5.16.2.49 `string Skylight.Physics.Config.ServerTypeServiceroom = "Lobby" + ServerTypeVersion` `[static]`

The service room

5.16.2.50 `string Skylight.Physics.Config.ServerTypeTrackingroom = "Tracking" + ServerTypeVersion` `[static]`

The tracking room

5.16.2.51 `string Skylight.Physics.Config.ServerTypeTutorialroom = "Tutorial" + ServerTypeVersion + "_world_"` `[static]`

The tutorial room

5.16.2.52 `int Skylight.Physics.Config.ServerTypeVersion = 176` `[static]`

The server version.

5.16.2.53 `int Skylight.Physics.Config.ShopPotionMax = 10` `[static]`

The shop_potion_max

5.16.2.54 `bool Skylight.Physics.Config.ShowBlacklistInvitation = false` `[static]`

The show_blacklist_invitation

5.16.2.55 `bool Skylight.Physics.Config.ShowDebugFriendrequest = false` `[static]`

The show_debug_friendrequest

5.16.2.56 `bool Skylight.Physics.Config.ShowDebugProfile = true` `[static]`

The show_debug_profile

5.16.2.57 `bool Skylight.Physics.Config.ShowDisabledShopitems = false` `[static]`

The show_disabled_shopitems

5.16.2.58 `string [] Skylight.Physics.Config.TutorialNames = {"Moving", "Gravity", "Edit"}` `[static]`

The tutorial_names

5.16.2.59 `string Skylight.Physics.Config.UrlBlog = "http://blog.everybodyedits.com"` `[static]`

The url for the Everybody Edits blog

5.16.2.60 `string Skylight.Physics.Config.UrlClubmemberAboutPage = "http://everybodyedits.com/club"` `[static]`

The url to the club member page

5.16.2.61 `string Skylight.Physics.Config.UrlHelpPage = "http://everybodyedits.com/help"` `[static]`

The help website

5.16.2.62 `string Skylight.Physics.Config.UrlTermsPage = "http://everybodyedits.com/terms"` `[static]`

The terms of service website

5.16.2.63 `bool Skylight.Physics.Config.UseDebugServer = false` `[static]`

Whether or not to use the debug server

5.16.2.64 `int Skylight.Physics.Config.Width = 640` `[static]`

The width

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

- Skylight/Physics/Config.cs

5.17 Skylight.Miscellaneous.SmileyIds.Contest Class Reference

Class [Contest](#).

Public Attributes

- `const int Bird = 29`

The types of smilies available for contest winners.

5.17.1 Detailed Description

Class [Contest](#).

5.17.2 Member Data Documentation

5.17.2.1 `const int Skylight.Miscellaneous.SmileyIds.Contest.Bird = 29`

The types of smilies available for contest winners.

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

- Skylight/Miscellaneous/SmileyIds.cs

5.18 Skylight.Crown Class Reference

Public Member Functions

- **Crown** ([In](#) @in)
- void **OnCrown** (Message m)

Events

- [In.PlayerEvent CrownEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.18.1 Event Documentation

5.18.1.1 [In.PlayerEvent Skylight.Crown.CrownEvent](#) = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Crown.cs

5.19 Skylight.FaceChange Class Reference

Public Member Functions

- **FaceChange** ([In](#) @in)
- void **OnFace** (Message m)

Events

- [In.PlayerEvent FaceEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.19.1 Event Documentation

5.19.1.1 [In.PlayerEvent Skylight.FaceChange.FaceEvent](#) = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/FaceChange.cs

5.20 Skylight.Miscellaneous.SmileyIds.Free Class Reference

Class [Free](#).

Public Attributes

- const int [Smile](#) = 0

The smiles that are included by default.

5.20.1 Detailed Description

Class [Free](#).

5.20.2 Member Data Documentation

5.20.2.1 const int Skylight.Miscellaneous.SmileyIds.Free.Smile = 0

The smiles that are included by default.

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

- Skylight/Miscellaneous/SmileyIds.cs

5.21 Skylight.GetWoot Class Reference

Public Member Functions

- **GetWoot** ([In](#) @in)
- void **OnGetWoot** (Message m)

Events

- [In.PlayerEvent MagicCoinEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.21.1 Event Documentation

5.21.1.1 In.PlayerEvent Skylight.GetWoot.MagicCoinEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/GetWoot.cs

5.22 Skylight.GiveWizard2 Class Reference

Public Member Functions

- **GiveWizard2** ([In](#) @in)
- void **OnGiveWizard2** (Message m)

Events

- [In.PlayerEvent](#) [RedWizardEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.22.1 Event Documentation

5.22.1.1 [In.PlayerEvent](#) Skylight.GiveWizard2.RedWizardEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/GiveWizard2.cs

5.23 Skylight.GodMode Class Reference

Public Member Functions

- **GodMode** ([In](#) @in)
- void **OnGod** (Message m)

Events

- [In.PlayerEvent](#) [GodEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.23.1 Event Documentation

5.23.1.1 [In.PlayerEvent](#) Skylight.GodMode.GodEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/GodMode.cs

5.24 Skylight.Grinch Class Reference

Public Member Functions

- **Grinch** ([In](#) @in)
- void **OnGiveGrinch** (Message m)

Events

- [In.PlayerEvent](#) [GrinchEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.24.1 Event Documentation

5.24.1.1 [In.PlayerEvent](#) Skylight.Grinch.GrinchEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Grinch.cs

5.25 Skylight.Hide Class Reference

Public Member Functions

- **Hide** ([In](#) @in)
- void **OnHide** ()

Events

- [In.RoomEvent](#) [HideEvent](#) = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.25.1 Event Documentation

5.25.1.1 [In.RoomEvent](#) Skylight.Hide.HideEvent = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/Hide.cs

5.26 Skylight.In Class Reference

The main class that takes in events from the playerio client.

Public Member Functions

- delegate void [BlockEvent](#) ([BlockEventArgs](#) e)
The block event compiled from the message from the server.
- delegate void [ChatEvent](#) ([ChatEventArgs](#) e)
A chat event (when the player sends a message).
- delegate void [PlayerEvent](#) ([PlayerEventArgs](#) e)
An event that concerns the player.
- delegate void [RoomEvent](#) ([RoomEventArgs](#) e)
Something changed in the room (for example the title).

Protected Member Functions

- virtual void [OnSignBlockEvent](#) (Message m)
When a sign block is placed in the world.

Properties

- [Bot](#) **Bot** [get, set]
- [Room](#) **Source** [get, set]
- [Add](#) **Add** [get]
- [Potions](#) **Potions** [get]
- [Autotext](#) **Autotext** [get]
- [BlockChanged](#) **BlockChanged** [get]
- [CoinObject](#) **CoinObject** [get]
- [AddSpecialBlock](#) **AddSpecialBlock** [get]
- [NoteBlock](#) **NoteBlock** [get]
- [OnCoinGet](#) **OnCoinGet** [get]
- [ClearMap](#) **ClearMap** [get]
- [FaceChange](#) **FaceChange** [get]
- [Grinch](#) **Grinch** [get]
- [Witch](#) **Witch** [get]
- [Wizard](#) **Wizard** [get]
- [GiveWizard2](#) **GiveWizard2** [get]
- [GodMode](#) **GodMode** [get]
- [Hide](#) **Hide** [get]
- [Crown](#) **Crown** [get]
- [OnKill](#) **OnKill1** [get]
- [Trophy](#) **Trophy** [get]
- [SignBlock](#) **SignBlock** [get]
- [LeftWorld](#) **LeftWorld** [get]
- [LevelChange](#) **LevelChange** [get]
- [Move](#) **Move** [get]
- [Moderator](#) **Moderator** [get]
- [Potion](#) **Potion** [get]
- [RefreshShop](#) **RefreshShop** [get]
- [ResetWorld](#) **ResetWorld** [get]
- [Chat](#) **Chat** [get]

- [ChatOld](#) **ChatOld** [get]
- [Save](#) **Save** [get]
- [Show](#) **Show** [get]
- [Tele](#) **Tele** [get]
- [Teleport](#) **Teleport** [get]
- [Meta](#) **Meta** [get]
- [Upgrade](#) **Upgrade** [get]
- [Wp](#) **Wp** [get]
- [Write](#) **Write** [get]
- [GetWoot](#) **GetWoot** [get]
- [WootUp](#) **WootUp** [get]
- [Access](#) **Access** [get]
- [Info](#) **Info** [get]

Events

- [BlockEvent](#) **CoinBlockEvent** = delegate { }
All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).
- [PlayerEvent](#) **AddEvent** = delegate { }
*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*
- [RoomEvent](#) **InitEvent** = delegate { }
Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.26.1 Detailed Description

The main class that takes in events from the playerio client.

5.26.2 Member Function Documentation

5.26.2.1 `delegate void Skylight.In.BlockEvent (BlockEventArgs e)`

The block event compiled from the message from the server.

Parameters

<i>e</i>	The block object.
----------	-------------------

5.26.2.2 `delegate void Skylight.In.ChatEvent (ChatEventArgs e)`

A chat event (when the player sends a message).

Parameters

<i>e</i>	The ChatEventArgs event.
----------	--------------------------

5.26.2.3 `virtual void Skylight.In.OnSignBlockEvent (Message m)` [protected],[virtual]

When a sign block is placed in the world.

5.26.2.4 `delegate void Skylight.In.PlayerEvent (PlayerEventArgs e)`

An event that concerns the player.

Parameters

<i>e</i>	The player object.
----------	--------------------

5.26.2.5 `delegate void Skylight.In.RoomEvent (RoomEventArgs e)`

Something changed in the room (for example the title).

Parameters

<i>e</i>	The room object.
----------	------------------

5.26.3 Event Documentation

5.26.3.1 `PlayerEvent Skylight.In.AddEvent = delegate { }`

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

5.26.3.2 `BlockEvent Skylight.In.CoinBlockEvent = delegate { }`

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

5.26.3.3 `RoomEvent Skylight.In.InitEvent = delegate { }`

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/In.cs

5.27 Skylight.Info Class Reference

Public Member Functions

- **Info** ([In](#) @in)
- void **OnInfo** (Message m)

Events

- [In.PlayerEvent InfoEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.27.1 Event Documentation

5.27.1.1 In.PlayerEvent Skylight.Info.InfoEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

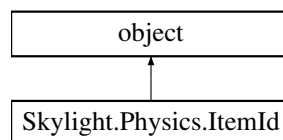
The documentation for this class was generated from the following file:

- Skylight/Info.cs

5.28 Skylight.Physics.ItemId Class Reference

Class item id.

Inheritance diagram for Skylight.Physics.ItemId:



Static Public Member Functions

- static bool [IsSolid](#) (int param1)
Determines whether the specified block is solid.
- static bool [IsClimbable](#) (int param1)
Determines whether the specified block is climbable.
- static bool [IsBackgroundRotateable](#) (int param1)
Determines whether [is background rotateable] [the specified block].
- static bool [IsDecorationRotateable](#) (int param1)
Determines whether [is decoration rotateable] [the specified block].

5.28.1 Detailed Description

Class item id.

5.28.2 Member Function Documentation

5.28.2.1 static bool Skylight.Physics.ItemId.IsBackgroundRotateable (int *param1*) [static]

Determines whether [is background rotateable] [the specified block].

Parameters

<i>param1</i>	The param1.
---------------	-------------

Returns

true if [is background rotateable] [the specified block]; otherwise, false.

5.28.2.2 static bool Skylight.Physics.ItemId.IsClimbable (int *param1*) [static]

Determines whether the specified block is climbable.

Parameters

<i>param1</i>	The block.
---------------	------------

Returns

`true` if the specified block is climbable; otherwise, `false`.

5.28.2.3 `static bool Skylight.Physics.ItemId.IsDecorationRotateable (int param1) [static]`

Determines whether [is decoration rotateable] [the specified block].

Parameters

<i>param1</i>	The param1.
---------------	-------------

Returns

`true` if [is decoration rotateable] [the specified block]; otherwise, `false`.

5.28.2.4 `static bool Skylight.Physics.ItemId.IsSolid (int param1) [static]`

Determines whether the specified block is solid.

Parameters

<i>param1</i>	The param1.
---------------	-------------

Returns

`true` if the specified block is solid; otherwise, `false`.

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

- Skylight/Physics/ItemId.cs

5.29 Skylight.LeftWorld Class Reference

Public Member Functions

- **LeftWorld** ([In](#) @in)
- void **OnLeft** (Message m)

Events

- [In.PlayerEvent LeaveEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.29.1 Event Documentation

5.29.1.1 In.PlayerEvent Skylight.LeftWorld.LeaveEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/LeftWorld.cs

5.30 Skylight.LevelChange Class Reference

Public Member Functions

- **LevelChange** (In @in)
- void **OnLevelUp** (Message m)

Events

- In.PlayerEvent LevelUpEvent = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.30.1 Event Documentation

5.30.1.1 In.PlayerEvent Skylight.LevelChange.LevelUpEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/LevelChange.cs

5.31 Skylight.Meta Class Reference

Public Member Functions

- **Meta** (In @in)
- void **OnUpdateMeta** (Message m)

Events

- In.RoomEvent UpdateMetaEvent = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.31.1 Event Documentation

5.31.1.1 In.RoomEvent Skylight.Meta.UpdateMetaEvent = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/Meta.cs

5.32 Skylight.Moderator Class Reference

Public Member Functions

- **Moderator** (In @in)
- void **OnMod** (Message m)

Events

- In.PlayerEvent ModModeEvent = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.32.1 Event Documentation

5.32.1.1 In.PlayerEvent Skylight.Moderator.ModModeEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Moderator.cs

5.33 Skylight.Move Class Reference

Public Member Functions

- **Move** (In @in)
- void **OnMove** (Message m)

Events

- In.PlayerEvent JumpEvent = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

- In.PlayerEvent MovementEvent = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.33.1 Event Documentation

5.33.1.1 `In.PlayerEvent Skylight.Move.JumpEvent = delegate { }`

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with `*SYSTEM`.

5.33.1.2 `In.PlayerEvent Skylight.Move.MovementEvent = delegate { }`

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with `*SYSTEM`.

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

- `Skylight/Move.cs`

5.34 Skylight.NoteBlock Class Reference

Public Member Functions

- **NoteBlock** (`In @in`)
- void **OnAddNoteblock** (`Message m`)

Events

- `In.BlockEvent SoundBlockEvent = delegate { }`

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

5.34.1 Event Documentation

5.34.1.1 `In.BlockEvent Skylight.NoteBlock.SoundBlockEvent = delegate { }`

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

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

- `Skylight/NoteBlock.cs`

5.35 Skylight.OnCoinGet Class Reference

Public Member Functions

- **OnCoinGet** (`In @in`)
- void **OnCoin** (`Message m`)

Events

- `In.PlayerEvent CoinCollectedEvent = delegate { }`

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with `*SYSTEM`.*

5.35.1 Event Documentation

5.35.1.1 In.PlayerEvent Skylight.OnCoinGet.CoinCollectedEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/OnCoinGet.cs

5.36 Skylight.OnKill Class Reference

Public Member Functions

- **OnKill** (In @in)
- void **OnKillPlayer** (Message m)

Events

- In.PlayerEvent **DeathEvent** = delegate { }

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

- Skylight/OnKill.cs

5.37 Skylight.Out Class Reference

Class **Out**. This class sends information to the server.

Public Member Functions

- void **Build** (int id, int x, int y)
Builds the specified block.
- void **Build** (Block b)
Builds the specified block object.
- void **Build** (List< Block > blockList)
Builds the specified block list.
- void **Clear** ()
Clears the entire world.
- void **HoldDown** (double startX, double startY)
Holds the down arrow key, causing the bot to move downward.
- void **HoldLeft** (double startX, double startY)
Holds the left arrow key.
- void **HoldRight** (double startX, double startY)
Holds the right arrow key.
- void **HoldUp** (double startX, double startY)
Holds the up arrow key.
- void **InputCode** (string editKey)

- Inputs the edit key.*
 - void [SetCode](#) (string newCode)
- Sets the edit key for the current room.*
 - void [Jump](#) (double startX, double startY)
- Tells the bot to jump from the specified coordinates.*
 - void [Move](#) (object[] args)
- Moves the specified bot.*
 - void [Move](#) (Message m)
- Overload for [Move](#). Moves using the Message object.*
 - void [Release](#) (double startX, double startY)
- Releases the arrow key.*
 - void [Say](#) (string s, bool useChatPrefix=true)
- Says the specified message.*
 - void [SetTitle](#) (string s)
- Sets the title of the room.*
 - void [Kick](#) (string name, string reason="")
- Kicks the specified player by their username.*
 - void [Kick](#) ([Player](#) p, string reason="")
- Kicks the specified [Player](#) object.*
 - void [Loadlevel](#) ()
- Resets the level to its state when it was last saved.*
 - void [Respawn](#) (string name)
- Respawns the specified player by their username.*
 - void [Respawn](#) ([Player](#) p)
- Respawns the specified [Player](#) object.*
 - void [RespawnAll](#) ()
- Respawns everyone in the room.*
 - void [Reset](#) ()
- Clears the entire world.*
 - void [Save](#) ()
- Saves the world.*
 - void [SetAllPotionBans](#) (bool value)
- Toggle all potion bans.*
 - void [SetEdit](#) (string name, bool value)
- Sets the edit of a player.*
 - void [SetEdit](#) ([Player](#) p, bool value)
- Sets the edit for a [Player](#) object.*
 - void [SetGod](#) (bool value)
- Sets the god mode for the bot.*
 - void [SetMute](#) (string name, bool value)
- Sets the mute for a player by their username. This will prevent chat messages from being sent from that player to the server.*
 - void [SetMute](#) ([Player](#) p, bool value)
- Sets the mute for a [Player](#) object.*
 - void [SetPotionBan](#) (int potionId, bool value)
- Sets the potion ban.*
 - void [SetSmiley](#) (int smileyId)
- Sets the smiley.*
 - void [SetVisibility](#) (bool value)
- Sets the visibility of the bot.*
 - void [Teleport](#) (int newXLocation, int newYLocation, string name="")

Teleports the specified new x location.

- void **Teleport** (int newXLocation, int newYLocation, **Player** p=null)

Teleports the specified new x location.

- void **TeleportAll** (int newXLocation, int newYLocation)

Teleports all.

Properties

- **Bot Bot** [get]

Gets the bot. This is what the developer will instantiate when they need to communicate with the bot.

- **Connection C** [get]

Gets the connection.

- **Room R** [get]

Gets the room.

5.37.1 Detailed Description

Class **Out**. This class sends information to the server.

5.37.2 Member Function Documentation

5.37.2.1 void Skylight.Out.Build (int *id*, int *x*, int *y*)

Builds the specified block.

Parameters

<i>id</i>	The id of the block.
<i>x</i>	The x coordinate of the block (in block units).
<i>y</i>	The y coordinate of the block (in block units).

5.37.2.2 void Skylight.Out.Build (**Block** *b*)

Builds the specified block object.

Parameters

<i>b</i>	The block.
----------	------------

5.37.2.3 void Skylight.Out.Build (List< **Block** > *blockList*)

Builds the specified block list.

Parameters

<i>blockList</i>	The block list.
------------------	-----------------

5.37.2.4 void Skylight.Out.Clear ()

Clears the entire world.

5.37.2.5 void Skylight.Out.HoldDown (double *startX*, double *startY*)

Holds the down arrow key, causing the bot to move downward.

Parameters

<i>startX</i>	The start x coordinate.
<i>startY</i>	The start y coordinate.

5.37.2.6 void Skylight.Out.HoldLeft (double *startX*, double *startY*)

Holds the left arrow key.

Parameters

<i>startX</i>	The start x coordinate.
<i>startY</i>	The start y coordinate.

5.37.2.7 void Skylight.Out.HoldRight (double *startX*, double *startY*)

Holds the right arrow key.

Parameters

<i>startX</i>	The start x coordinate.
<i>startY</i>	The start y coordinate.

5.37.2.8 void Skylight.Out.HoldUp (double *startX*, double *startY*)

Holds the up arrow key.

Parameters

<i>startX</i>	The start x coordinate.
<i>startY</i>	The start y coordinate.

5.37.2.9 void Skylight.Out.InputCode (string *editKey*)

Inputs the edit key.

Parameters

<i>editKey</i>	The edit key.
----------------	---------------

5.37.2.10 void Skylight.Out.Jump (double *startX*, double *startY*)

Tells the bot to jump from the specified coordinates.

Parameters

<i>startX</i>	The start x coordinate.
<i>startY</i>	The start y coordinate.

5.37.2.11 void Skylight.Out.Kick (string *name*, string *reason* = " ")

Kicks the specified player by their username.

Parameters

<i>name</i>	The username.
<i>reason</i>	The reason.

5.37.2.12 void Skylight.Out.Kick (Player *p*, string *reason* = " ")

Kicks the specified [Player](#) object.

Parameters

<i>p</i>	The player object.
<i>reason</i>	The reason.

5.37.2.13 void Skylight.Out.Loadlevel ()

Resets the level to its state when it was last saved.

5.37.2.14 void Skylight.Out.Move (object[] *args*)

Moves the specified bot.

Parameters

<i>args</i>	The raw message where to move.
-------------	--------------------------------

5.37.2.15 void Skylight.Out.Move (Message *m*)

Overload for [Move](#). Moves using the Message object.

Parameters

<i>m</i>	The movement Message Object.
----------	------------------------------

5.37.2.16 void Skylight.Out.Release (double *startX*, double *startY*)

Releases the arrow key.

Parameters

<i>startX</i>	The start x.
<i>startY</i>	The start y.

5.37.2.17 void Skylight.Out.Reset ()

Clears the entire world.

5.37.2.18 void Skylight.Out.Respawn (string *name*)

Respawns the specified player by their username.

Parameters

<i>name</i>	The username.
-------------	---------------

5.37.2.19 void Skylight.Out.Respawn (Player *p*)

Respawns the specified [Player](#) object.

Parameters

<i>p</i>	The player.
----------	-------------

5.37.2.20 void Skylight.Out.RespawnAll ()

Respawns everyone in the room.

5.37.2.21 void Skylight.Out.Save ()

Saves the world.

5.37.2.22 void Skylight.Out.Say (string *s*, bool *useChatPrefix* = true)

Says the specified message.

Parameters

<i>s</i>	The message.
<i>useChatPrefix</i>	if set to <code>true</code> then [use chat prefix].

5.37.2.23 void Skylight.Out.SetAllPotionBans (bool *value*)

Toggle all potion bans.

Parameters

<i>value</i>	if set to <code>true</code> then turn on potions.
--------------	---

5.37.2.24 void Skylight.Out.SetCode (string *newCode*)

Sets the edit key for the current room.

Parameters

<i>newCode</i>	The new code.
----------------	---------------

5.37.2.25 void Skylight.Out.SetEdit (string *name*, bool *value*)

Sets the edit of a player.

Parameters

<i>name</i>	The username of the player.
<i>value</i>	if set to <code>true</code> then the player will receive edit privileges.

5.37.2.26 void Skylight.Out.SetEdit (Player *p*, bool *value*)

Sets the edit for a [Player](#) object.

Parameters

<i>p</i>	The Player object.
<i>value</i>	if set to <code>true</code> then the Player object receives edit.

5.37.2.27 void Skylight.Out.SetGod (bool *value*)

Sets the god mode for the bot.

Parameters

<i>value</i>	if set to <code>true</code> then the bot will go into god mode.
--------------	---

5.37.2.28 void Skylight.Out.SetMute (string *name*, bool *value*)

Sets the mute for a player by their username. This will prevent chat messages from being sent from that player to the server.

Parameters

<i>name</i>	The username.
<i>value</i>	if set to <code>true</code> then that username will be muted.

5.37.2.29 void Skylight.Out.SetMute (Player *p*, bool *value*)

Sets the mute for a [Player](#) object.

Parameters

<i>p</i>	The Player .
<i>value</i>	if set to <code>true</code> then that <code>Player.subject</code> object will be muted.

5.37.2.30 void Skylight.Out.SetPotionBan (int *potionId*, bool *value*)

Sets the potion ban.

Parameters

<i>potionId</i>	The potion id.
<i>value</i>	if set to <code>true</code> then potions will be turned on for that potion.

5.37.2.31 void Skylight.Out.SetSmiley (int *smileyId*)

Sets the smiley.

Parameters

<i>smileyId</i>	The smiley id.
-----------------	----------------

5.37.2.32 void Skylight.Out.SetTitle (string s)

Sets the title of the room.

Parameters

<i>s</i>	The new title.
----------	----------------

5.37.2.33 void Skylight.Out.SetVisibility (bool value)

Sets the visibility of the bot.

Parameters

<i>value</i>	if set to <code>true</code> then the bot will become visible.
--------------	---

5.37.2.34 void Skylight.Out.Teleport (int newXLocation, int newYLocation, string name = " ")

Teleports the specified new x location.

Parameters

<i>newXLocation</i>	The new x location.
<i>newYLocation</i>	The new y location.
<i>name</i>	The name.

5.37.2.35 void Skylight.Out.Teleport (int newXLocation, int newYLocation, Player p = null)

Teleports the specified new x location.

Parameters

<i>newXLocation</i>	The new x location.
<i>newYLocation</i>	The new y location.
<i>p</i>	The p.

5.37.2.36 void Skylight.Out.TeleportAll (int newXLocation, int newYLocation)

Teleports all.

Parameters

<i>newXLocation</i>	The new x location.
<i>newYLocation</i>	The new y location.

5.37.3 Property Documentation

5.37.3.1 Bot Skylight.Out.Bot [get]

Gets the bot. This is what the developer will instantiate when they need to communicate with the bot.

The bot.

5.37.3.2 Connection Skylight.Out.C [get]

Gets the connection.

The current connection.

5.37.3.3 Room Skylight.Out.R [get]

Gets the room.

The room.

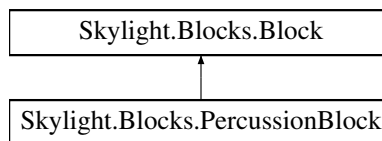
The documentation for this class was generated from the following file:

- Skylight/Out.cs

5.38 Skylight.Blocks.PercussionBlock Class Reference

Class [PercussionBlock](#).

Inheritance diagram for Skylight.Blocks.PercussionBlock:



Public Member Functions

- [PercussionBlock](#) (int x, int y, int percussionId)
Initializes a new instance of the [PercussionBlock](#) class.

Public Attributes

- const int [Base1](#) = 0
The types of notes that can be played.

Properties

- int [PercussionId](#) [get, set]
Gets the percussion identifier.

5.38.1 Detailed Description

Class [PercussionBlock](#).

5.38.2 Constructor & Destructor Documentation

5.38.2.1 Skylight.Blocks.PercussionBlock.PercussionBlock (int *x*, int *y*, int *percussionId*)

Initializes a new instance of the [PercussionBlock](#) class.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.
<i>percussionId</i>	The percussion identifier.

5.38.3 Member Data Documentation

5.38.3.1 `const int Skylight.Blocks.PercussionBlock.Base1 = 0`

The types of notes that can be played.

5.38.4 Property Documentation

5.38.4.1 `int Skylight.Blocks.PercussionBlock.PercussionId` `[get]`, `[set]`

Gets the percussion identifier.

The percussion identifier.

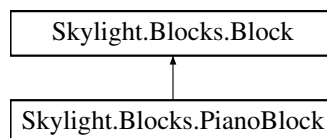
The documentation for this class was generated from the following file:

- `Skylight/Blocks/PercussionBlock.cs`

5.39 Skylight.Blocks.PianoBlock Class Reference

Class [PianoBlock](#).

Inheritance diagram for `Skylight.Blocks.PianoBlock`:



Public Member Functions

- [PianoBlock](#) (int x, int y, int key)
Initializes a new instance of the [PianoBlock](#) class.

Public Attributes

- `const int C1 = 1`
The piano notes.

Properties

- `int Pianoid` `[get]`, `[set]`
Gets the piano identifier.

5.39.1 Detailed Description

Class [PianoBlock](#).

5.39.2 Constructor & Destructor Documentation

5.39.2.1 Skylight.Blocks.PianoBlock.PianoBlock (int x, int y, int key)

Initializes a new instance of the [PianoBlock](#) class.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.
<i>key</i>	The key (note).

5.39.3 Member Data Documentation

5.39.3.1 const int Skylight.Blocks.PianoBlock.C1 = 1

The piano notes.

5.39.4 Property Documentation

5.39.4.1 int Skylight.Blocks.PianoBlock.Pianoid [get], [set]

Gets the piano identifier.

The piano identifier.

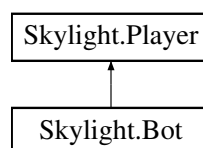
The documentation for this class was generated from the following file:

- Skylight/Blocks/PianoBlock.cs

5.40 Skylight.Player Class Reference

Class [Player](#).

Inheritance diagram for Skylight.Player:



Public Member Functions

- [Player](#) ([Room](#) room, int id, string name, int smiley, double xPos, double yPos, bool isGod, bool isMod, bool hasChat, int coins, bool purple, bool isFriend, int level, bool hasClub, bool isInvulnerable, bool isThrusting, bool isZombie, bool isDead, bool levitation)

Initializes a new instance of the [Player](#) class.

- void [Tick](#) ()

Ticks this instance.

Public Attributes

- int [CheckpointX](#) = -1
The checkpoint at the x coordinate
- int [CheckpointY](#) = -1
The checkpoint at the y coordinate
- int [CurrentBlockId](#) = 0
The current block identifier
- bool [Isclubmember](#) = false
Whether the player is a club member or not
- bool [SwitchOpened](#) = false
The switch opened
- double [X](#) = 0
The x position
- double [Y](#) = 0
The y position

Static Public Attributes

- static readonly List< string > [Admins](#)
The admins. This is never used.

Protected Attributes

- double [Boost](#)
The boost
- double [Gravity](#)
The gravity
- double [ModifierX](#) = 0
The modifier x
- double [ModifierY](#) = 0
The modifier y
- double [MudBuoyancy](#)
The mud buoyancy
- double [MudDrag](#)
The mud drag
- double [NoModifierDragX](#)
The no modifier drag x
- double [NoModifierDragY](#)
The no modifier drag y
- int [Size](#)
The size of the block
- double [SpeedX](#) = 0
The speed x
- double [SpeedY](#) = 0
The speed y
- double [WaterBuoyancy](#)
The water buoyancy
- double [WaterDrag](#)
The water drag

Properties

- bool [HasAccess](#) [get, set]
Gets a value indicating whether this instance has access.
- bool [HasBoost](#) [get, set]
Gets a value indicating whether this instance has boost.
- bool [HasClub](#) [get, set]
Gets a value indicating whether this instance has club.
- bool [HasCommandAccess](#) [get, set]
Gets a value indicating whether this instance has command access.
- bool [HasCrown](#) [get, set]
Gets a value indicating whether this instance has crown.
- bool [HasGravityModifier](#) [get, set]
Gets a value indicating whether this instance has gravity modifier.
- bool [HasSilverCrown](#) [get, set]
Gets a value indicating whether this instance has silver crown.
- bool [IsBot](#) [get]
Gets a value indicating whether this instance is bot.
- bool [IsFriend](#) [get, set]
Gets a value indicating whether this instance is friend.
- int [Level](#) [get, set]
The level of the player (in terms of xp).
- bool [IsGod](#) [get, set]
Gets a value indicating whether this instance is god.
- bool [IsMod](#) [get, set]
Gets a value indicating whether this instance is mod.
- bool [HasChat](#) [get, set]
Whether or not the player is able to use the chat box for free form chat messages.
- bool [IsHoldingLeft](#) [get, set]
Gets a value indicating whether this instance is holding left.
- bool [IsHoldingRight](#) [get, set]
Gets a value indicating whether this instance is holding right.
- bool [IsHoldingUp](#) [get, set]
Gets a value indicating whether this instance is holding up.
- bool [IsHoldingDown](#) [get, set]
Gets a value indicating whether this instance is holding down.
- bool [IsHoldingSpace](#) [get, set]
Gets a value indicating whether this instance is holding space.
- bool [IsOwner](#) [get, set]
Gets a value indicating whether this instance is owner.
- int [Coins](#) [get, set]
Gets the coins.
- int [BlueCoins](#) [get, set]
Gets the blue coins.
- int [CollectedMagic](#) [get, set]
Gets the collected magic.
- int [DeathCount](#) [get, set]
Gets the death count.
- int [Id](#) [get, set]
Gets the identifier.
- int [Smiley](#) [get, set]

- Gets the smiley.*
- int [XpLevel](#) [get, set]
Gets the xp level.
- List< int > [PotionEffects](#) [get, set]
Gets the potion effects.
- [Room](#) [PlayingIn](#) [get, set]
Gets the playing in.
- string [Name](#) [get, set]
Gets the name.
- int [BlockX](#) [get, set]
Gets or sets the block x.
- int [BlockY](#) [get, set]
Gets or sets the block y.

5.40.1 Detailed Description

Class [Player](#).

5.40.2 Constructor & Destructor Documentation

- 5.40.2.1 **Skylight.Player.Player ([Room](#) *room*, int *id*, string *name*, int *smiley*, double *xPos*, double *yPos*, bool *isGod*, bool *isMod*, bool *hasChat*, int *coins*, bool *purple*, bool *isFriend*, int *level*, bool *hasClub*, bool *isInvulnerable*, bool *isThrusting*, bool *isZombie*, bool *isDead*, bool *levitation*)**

Initializes a new instance of the [Player](#) class.

Parameters

<i>room</i>	The room.
<i>id</i>	The identifier.
<i>name</i>	The name.
<i>smiley</i>	The smiley.
<i>xPos</i>	The x position.
<i>yPos</i>	The y position.
<i>isGod</i>	if set to <code>true</code> [is god].
<i>isMod</i>	if set to <code>true</code> [is mod].
<i>hasChat</i>	if set to <code>true</code> [has chat].
<i>coins</i>	The coins.
<i>purple</i>	if set to <code>true</code> [purple].
<i>isFriend</i>	if set to <code>true</code> [is friend].
<i>level</i>	The level.
<i>hasClub</i>	If the player has builder's club or not.
<i>isInvulnerable</i>	The player can die.
<i>isThrusting</i>	Player is using boost potion
<i>isZombie</i>	Player is a zombie
<i>isDead</i>	Player is dead
<i>levitation</i>	Player has levitation potion

5.40.3 Member Function Documentation

- 5.40.3.1 **void Skylight.Player.Tick ()**

Ticks this instance.

5.40.4 Member Data Documentation

5.40.4.1 readonly List<string> Skylight.Player.Admins [static]

Initial value:

```
= new List<string>
{
    "benjaminsen",
    "cyclone",
    "toby",
    "rpgmaster2000",
    "mrshoe",
    "mrvoid"
}
```

The admins. This is never used.

5.40.4.2 double Skylight.Player.Boost [protected]

The boost

5.40.4.3 int Skylight.Player.CheckpointX = -1

The checkpoint at the x coordinate

5.40.4.4 int Skylight.Player.CheckpointY = -1

The checkpoint at the y coordinate

5.40.4.5 int Skylight.Player.CurrentBlockId = 0

The current block identifier

5.40.4.6 double Skylight.Player.Gravity [protected]

The gravity

5.40.4.7 bool Skylight.Player.Isclubmember = false

Whether the player is a club member or not

5.40.4.8 double Skylight.Player.ModifierX = 0 [protected]

The modifier x

5.40.4.9 double Skylight.Player.ModifierY = 0 [protected]

The modifier y

5.40.4.10 double Skylight.Player.MudBuoyancy [protected]

The mud buoyancy

5.40.4.11 `double Skylight.Player.MudDrag` `[protected]`

The mud drag

5.40.4.12 `double Skylight.Player.NoModifierDragX` `[protected]`

The no modifier drag x

5.40.4.13 `double Skylight.Player.NoModifierDragY` `[protected]`

The no modifier drag y

5.40.4.14 `int Skylight.Player.Size` `[protected]`

The size of the block

5.40.4.15 `double Skylight.Player.SpeedX = 0` `[protected]`

The speed x

5.40.4.16 `double Skylight.Player.SpeedY = 0` `[protected]`

The speed y

5.40.4.17 `bool Skylight.Player.SwitchOpened = false`

The switch opened

5.40.4.18 `double Skylight.Player.WaterBuoyancy` `[protected]`

The water buoyancy

5.40.4.19 `double Skylight.Player.WaterDrag` `[protected]`

The water drag

5.40.4.20 `double Skylight.Player.X = 0`

The x position

5.40.4.21 `double Skylight.Player.Y = 0`

The y position

5.40.5 Property Documentation

5.40.5.1 `int Skylight.Player.BlockX` `[get]`, `[set]`

Gets or sets the block x.

The block x.

5.40.5.2 `int Skylight.Player.BlockY` `[get], [set]`

Gets or sets the block y.

The block y.

5.40.5.3 `int Skylight.Player.BlueCoins` `[get], [set]`

Gets the blue coins.

The blue coins.

5.40.5.4 `int Skylight.Player.Coins` `[get], [set]`

Gets the coins.

The coins.

5.40.5.5 `int Skylight.Player.CollectedException` `[get], [set]`

Gets the collected magic.

The collected magic.

5.40.5.6 `int Skylight.Player.DeathCount` `[get], [set]`

Gets the death count.

The death count.

5.40.5.7 `bool Skylight.Player.HasAccess` `[get], [set]`

Gets a value indicating whether this instance has access.

`true` if this instance has access; otherwise, `false`.

5.40.5.8 `bool Skylight.Player.HasBoost` `[get], [set]`

Gets a value indicating whether this instance has boost.

`true` if this instance has boost; otherwise, `false`.

5.40.5.9 `bool Skylight.Player.HasChat` `[get], [set]`

Whether or not the player is able to use the chat box for free form chat messages.

`true` if this instance has chat; otherwise, `false`.

5.40.5.10 `bool Skylight.Player.HasClub` `[get], [set]`

Gets a value indicating whether this instance has club.

`true` if this instance has club; otherwise, `false`.

5.40.5.11 bool Skylight.Player.HasCommandAccess [get], [set]

Gets a value indicating whether this instance has command access.

true if this instance has command access; otherwise, false.

5.40.5.12 bool Skylight.Player.HasCrown [get], [set]

Gets a value indicating whether this instance has crown.

true if this instance has crown; otherwise, false.

5.40.5.13 bool Skylight.Player.HasGravityModifier [get], [set]

Gets a value indicating whether this instance has gravity modifier.

true if this instance has gravity modifier; otherwise, false.

5.40.5.14 bool Skylight.Player.HasSilverCrown [get], [set]

Gets a value indicating whether this instance has silver crown.

true if this instance has silver crown; otherwise, false.

5.40.5.15 int Skylight.Player.Id [get], [set]

Gets the identifier.

The identifier.

5.40.5.16 bool Skylight.Player.IsBot [get]

Gets a value indicating whether this instance is bot.

true if this instance is bot; otherwise, false.

5.40.5.17 bool Skylight.Player.IsFriend [get], [set]

Gets a value indicating whether this instance is friend.

true if this instance is friend; otherwise, false.

5.40.5.18 bool Skylight.Player.IsGod [get], [set]

Gets a value indicating whether this instance is god.

true if this instance is god; otherwise, false.

5.40.5.19 bool Skylight.Player.IsHoldingDown [get], [set]

Gets a value indicating whether this instance is holding down.

true if this instance is holding down; otherwise, false.

5.40.5.20 `bool Skylight.Player.IsHoldingLeft` `[get]`, `[set]`

Gets a value indicating whether this instance is holding left.
`true` if this instance is holding left; otherwise, `false`.

5.40.5.21 `bool Skylight.Player.IsHoldingRight` `[get]`, `[set]`

Gets a value indicating whether this instance is holding right.
`true` if this instance is holding right; otherwise, `false`.

5.40.5.22 `bool Skylight.Player.IsHoldingSpace` `[get]`, `[set]`

Gets a value indicating whether this instance is holding space.
`true` if this instance is holding space; otherwise, `false`.

5.40.5.23 `bool Skylight.Player.IsHoldingUp` `[get]`, `[set]`

Gets a value indicating whether this instance is holding up.
`true` if this instance is holding up; otherwise, `false`.

5.40.5.24 `bool Skylight.Player.IsMod` `[get]`, `[set]`

Gets a value indicating whether this instance is mod.
`true` if this instance is mod; otherwise, `false`.

5.40.5.25 `bool Skylight.Player.IsOwner` `[get]`, `[set]`

Gets a value indicating whether this instance is owner.
`true` if this instance is owner; otherwise, `false`.

5.40.5.26 `int Skylight.Player.Level` `[get]`, `[set]`

The level of the player (in terms of xp).
The level.

5.40.5.27 `string Skylight.Player.Name` `[get]`, `[set]`

Gets the name.
The name.

5.40.5.28 `Room Skylight.Player.PlayingIn` `[get]`, `[set]`

Gets the playing in.
The playing in.

5.40.5.29 `List<int> Skylight.Player.PotionEffects` `[get]`, `[set]`

Gets the potion effects.

The potion effects.

5.40.5.30 `int Skylight.Player.Smiley` `[get]`, `[set]`

Gets the smiley.

The smiley.

5.40.5.31 `int Skylight.Player.XpLevel` `[get]`, `[set]`

Gets the xp level.

The xp level.

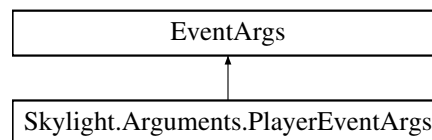
The documentation for this class was generated from the following file:

- `Skylight/Player.cs`

5.41 Skylight.Arguments.PlayerEventArgs Class Reference

This class sets the properties about a player such as who the player is (username), their most recent message and where they are (the room that they are in).

Inheritance diagram for `Skylight.Arguments.PlayerEventArgs`:



Public Member Functions

- `PlayerEventArgs` (`Player` subject, `Room` origin, `Message` rawMessage)
The main constructor method.

Properties

- `Player Subject` `[get]`, `[set]`
The username of the player.
- `Room Origin` `[get]`, `[set]`
The room that the player is originating in.
- `Message RawMessage` `[get]`, `[set]`
The raw, unparsed message from the server.

5.41.1 Detailed Description

This class sets the properties about a player such as who the player is (username), their most recent message and where they are (the room that they are in).

5.41.2 Constructor & Destructor Documentation

5.41.2.1 Skylight.Arguments.PlayerEventArgs.PlayerEventArgs (Player subject, Room origin, Message rawMessage)

The main constructor method.

Parameters

<i>subject</i>	The player
<i>origin</i>	The room where the player is originating in.
<i>rawMessage</i>	The raw, unparsed message from the server (concerning player).

5.41.3 Property Documentation

5.41.3.1 Room Skylight.Arguments.PlayerEventArgs.Origin [get], [set]

The room that the player is originating in.

5.41.3.2 Message Skylight.Arguments.PlayerEventArgs.RawMessage [get], [set]

The raw, unparsed message from the server.

5.41.3.3 Player Skylight.Arguments.PlayerEventArgs.Subject [get], [set]

The username of the player.

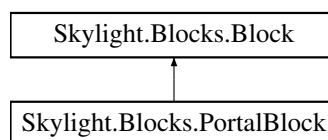
The documentation for this class was generated from the following file:

- Skylight/Arguments/PlayerEventArgs.cs

5.42 Skylight.Blocks.PortalBlock Class Reference

Class [PortalBlock](#).

Inheritance diagram for Skylight.Blocks.PortalBlock:



Public Member Functions

- [PortalBlock](#) (int x, int y, int direction, int portalId, int portalDestination, bool visible)
Initializes a new instance of the [PortalBlock](#) class.

Properties

- new int [Direction](#) [get, set]
The direction of the portal.
- int [PortalDestination](#) [get, set]

Gets the portal destination.

- int `PortalId` [get, set]

Gets the portal identifier.

- bool `Visible` [get, set]

Gets a value indicating whether this `PortalBlock` is visible.

Additional Inherited Members

5.42.1 Detailed Description

Class `PortalBlock`.

5.42.2 Constructor & Destructor Documentation

5.42.2.1 `Skylight.Blocks.PortalBlock.PortalBlock (int x, int y, int direction, int portalId, int portalDestination, bool visible)`

Initializes a new instance of the `PortalBlock` class.

Parameters

<code>x</code>	The x coordinate.
<code>y</code>	The y coordiante.
<code>direction</code>	The direction.
<code>portalId</code>	The portal identifier.
<code>portalDestination</code>	The portal destination.
<code>visible</code>	if set to <code>true</code> then the portal is active.

5.42.3 Property Documentation

5.42.3.1 `new int Skylight.Blocks.PortalBlock.Direction` [get], [set]

The direction of the portal.

5.42.3.2 `int Skylight.Blocks.PortalBlock.PortalDestination` [get], [set]

Gets the portal destination.

The portal destination.

5.42.3.3 `int Skylight.Blocks.PortalBlock.PortalId` [get], [set]

Gets the portal identifier.

The portal identifier.

5.42.3.4 `bool Skylight.Blocks.PortalBlock.Visible` [get], [set]

Gets a value indicating whether this `PortalBlock` is visible.

`true` if visible; otherwise, `false`.

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

- `Skylight/Blocks/PortalBlock.cs`

5.43 Skylight.Potion Class Reference

Public Member Functions

- **Potion** ([In](#) @in)
- void **OnP** (Message m)

Events

- [In.PlayerEvent](#) **PotionEvent** = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.43.1 Event Documentation

5.43.1.1 [In.PlayerEvent](#) Skylight.Potion.PotionEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Potion.cs

5.44 Skylight.Potions Class Reference

Public Member Functions

- **Potions** ([In](#) @in)
- void **OnAllowPotions** (Message m)

Events

- [In.RoomEvent](#) **PotionToggleEvent** = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.44.1 Event Documentation

5.44.1.1 [In.RoomEvent](#) Skylight.Potions.PotionToggleEvent = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/Potions.cs

5.45 Skylight.RefreshShop Class Reference

Public Member Functions

- **RefreshShop** ([In](#) @in)
- void **OnRefreshShop** ()

Events

- [In.RoomEvent RefreshshopEvent](#) = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.45.1 Event Documentation

5.45.1.1 [In.RoomEvent](#) Skylight.RefreshShop.RefreshshopEvent = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/RefreshShop.cs

5.46 Skylight.ResetWorld Class Reference

Public Member Functions

- **ResetWorld** ([In](#) @in)
- void **OnReset** (Message m)

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

- Skylight/ResetWorld.cs

5.47 Skylight.Room Class Reference

Class [Room](#).

Public Member Functions

- [Room](#) (string id)
Initializes a new instance of the [Room](#) class.

Properties

- static List< [Room](#) > [JoinedRooms](#) [get, set]
Gets the joined rooms.
- [Block](#)[,] [Map](#) [get, set]
Gets the map.

- bool [BlocksLoaded](#) [get, set]
Gets a value indicating whether [blocks are loaded].
- bool [HasPull](#) [get, set]
Gets a value indicating whether this instance has pull.
- bool [IsInitialized](#) [get, set]
Gets a value indicating whether this instance is initialized.
- bool [IsTutorialRoom](#) [get, set]
Gets a value indicating whether this instance is tutorial room.
- bool [PotionsAllowed](#) [get, set]
Gets a value indicating whether [potions allowed].
- bool [TimeDoorsVisible](#) [get, set]
Gets a value indicating whether [time doors visible].
- [Bot Receiver](#) [get, set]
Gets the receiver.
- double [GravityMultiplier](#) [get, set]
Gets the gravity multiplier.
- [In Pull](#) [get, set]
Gets the pull.
- int [Height](#) [get, set]
Gets the height.
- int [Plays](#) [get, set]
Gets the plays.
- int [TotalWoots](#) [get, set]
Gets the total woots.
- int [Width](#) [get, set]
Gets the width.
- int [Woots](#) [get, set]
Gets the woots.
- List< [In](#) > [Pulls](#) [get, set]
Gets the pulls.
- List< KeyValuePair< string, [Player](#) > > [ChatLog](#) [get, set]
Gets the chat log.
- List< [Player](#) > [OnlinePlayers](#) [get, set]
Gets the online players.
- List< [Bot](#) > [OnlineBots](#) [get, set]
Gets the online bots.
- [Player Owner](#) [get, set]
Gets the owner.
- string [EditKey](#) [get, set]
Gets the edit key.
- string [Id](#) [get, set]
Gets the identifier.
- string [Name](#) [get, set]
Gets the name.
- string [RoomKey](#) [get, set]
Gets the room key.
- bool [RedActivated](#) [get, set]
Gets or sets a value indicating whether [red activated].
- bool [GreenActivated](#) [get, set]
Gets or sets a value indicating whether [green activated].
- bool [BlueActivated](#) [get, set]
Gets or sets a value indicating whether [blue activated].

5.47.1 Detailed Description

Class [Room](#).

5.47.2 Constructor & Destructor Documentation

5.47.2.1 Skylight.Room.Room (string *id*)

Initializes a new instance of the [Room](#) class.

Parameters

<i>id</i>	The room identifier.
-----------	----------------------

5.47.3 Property Documentation

5.47.3.1 bool Skylight.Room.BlocksLoaded [get], [set]

Gets a value indicating whether [blocks are loaded].

true if [blocks loaded]; otherwise, false.

5.47.3.2 bool Skylight.Room.BlueActivated [get], [set]

Gets or sets a value indicating whether [blue activated].

true if [blue activated]; otherwise, false.

5.47.3.3 List<KeyValuePair<string, Player>> Skylight.Room.ChatLog [get], [set]

Gets the chat log.

The chat log.

5.47.3.4 string Skylight.Room.EditKey [get], [set]

Gets the edit key.

The edit key.

5.47.3.5 double Skylight.Room.GravityMultiplier [get], [set]

Gets the gravity multiplier.

The gravity multiplier.

5.47.3.6 bool Skylight.Room.GreenActivated [get], [set]

Gets or sets a value indicating whether [green activated].

true if [green activated]; otherwise, false.

5.47.3.7 bool Skylight.Room.HasPull [get], [set]

Gets a value indicating whether this instance has pull.

true if this instance has pull; otherwise, false.

5.47.3.8 `int Skylight.Room.Height` [get], [set]

Gets the height.

The height.

5.47.3.9 `string Skylight.Room.Id` [get], [set]

Gets the identifier.

The identifier.

5.47.3.10 `bool Skylight.Room.IsInitialized` [get], [set]

Gets a value indicating whether this instance is initialized.

true if this instance is initialized; otherwise, false.

5.47.3.11 `bool Skylight.Room.IsTutorialRoom` [get], [set]

Gets a value indicating whether this instance is tutorial room.

true if this instance is tutorial room; otherwise, false.

5.47.3.12 `List<Room> Skylight.Room.JoinedRooms` [static], [get], [set]

Gets the joined rooms.

The joined rooms.

5.47.3.13 `Block[,] Skylight.Room.Map` [get], [set]

Gets the map.

The map.

5.47.3.14 `string Skylight.Room.Name` [get], [set]

Gets the name.

The name.

5.47.3.15 `List<Bot> Skylight.Room.OnlineBots` [get], [set]

Gets the online bots.

The online bots.

5.47.3.16 `List<Player> Skylight.Room.OnlinePlayers` [get], [set]

Gets the online players.

The online players.

5.47.3.17 Player Skylight.Room.Owner [get], [set]

Gets the owner.

The owner.

5.47.3.18 int Skylight.Room.Plays [get], [set]

Gets the plays.

The plays.

5.47.3.19 bool Skylight.Room.PotionsAllowed [get], [set]

Gets a value indicating whether [potions allowed].

true if [potions allowed]; otherwise, false.

5.47.3.20 In Skylight.Room.Pull [get], [set]

Gets the pull.

The pull.

5.47.3.21 List<In> Skylight.Room.Pulls [get], [set]

Gets the pulls.

The pulls.

5.47.3.22 Bot Skylight.Room.Receiver [get], [set]

Gets the receiver.

The receiver.

5.47.3.23 bool Skylight.Room.RedActivated [get], [set]

Gets or sets a value indicating whether [red activated].

true if [red activated]; otherwise, false.

5.47.3.24 string Skylight.Room.RoomKey [get], [set]

Gets the room key.

The room key.

5.47.3.25 bool Skylight.Room.TimeDoorsVisible [get], [set]

Gets a value indicating whether [time doors visible].

true if [time doors visible]; otherwise, false.

5.47.3.26 `int Skylight.Room.TotalWoots` `[get]`, `[set]`

Gets the total woots.

The total woots.

5.47.3.27 `int Skylight.Room.Width` `[get]`, `[set]`

Gets the width.

The width.

5.47.3.28 `int Skylight.Room.Woots` `[get]`, `[set]`

Gets the woots.

The woots.

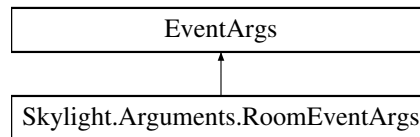
The documentation for this class was generated from the following file:

- Skylight/Room.cs

5.48 Skylight.Arguments.RoomEventArgs Class Reference

Class [RoomEventArgs](#).

Inheritance diagram for Skylight.Arguments.RoomEventArgs:



Public Member Functions

- [RoomEventArgs](#) ([Room](#) changedRoom)
Initializes a new instance of the [RoomEventArgs](#) class.

Properties

- [Room ChangedRoom](#) `[get]`
Gets the changed room.

5.48.1 Detailed Description

Class [RoomEventArgs](#).

5.48.2 Constructor & Destructor Documentation

5.48.2.1 `Skylight.Arguments.RoomEventArgs.RoomEventArgs (Room changedRoom)`

Initializes a new instance of the [RoomEventArgs](#) class.

Parameters

<i>changedRoom</i>	The changed room.
--------------------	-------------------

5.48.3 Property Documentation

5.48.3.1 Room Skylight.Arguments.RoomEventArgs.ChangedRoom [get]

Gets the changed room.

The changed room.

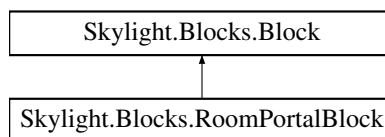
The documentation for this class was generated from the following file:

- Skylight/Arguments/RoomEventArgs.cs

5.49 Skylight.Blocks.RoomPortalBlock Class Reference

Class [RoomPortalBlock](#).

Inheritance diagram for Skylight.Blocks.RoomPortalBlock:



Public Member Functions

- [RoomPortalBlock](#) (int x, int y, string portalDestination)
Initializes a new instance of the [RoomPortalBlock](#) class.

Properties

- string [PortalDestination](#) [get, set]
Gets the portal destination.

Additional Inherited Members

5.49.1 Detailed Description

Class [RoomPortalBlock](#).

5.49.2 Constructor & Destructor Documentation

5.49.2.1 Skylight.Blocks.RoomPortalBlock.RoomPortalBlock (int x, int y, string portalDestination)

Initializes a new instance of the [RoomPortalBlock](#) class.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.
<i>portalDestination</i>	The portal destination.

5.49.3 Property Documentation

5.49.3.1 string Skylight.Blocks.RoomPortalBlock.PortalDestination [get], [set]

Gets the portal destination.

The portal destination.

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

- Skylight/Blocks/RoomPortalBlock.cs

5.50 Skylight.Save Class Reference

Public Member Functions

- **Save** ([In](#) @in)
- void **OnSaved** ()

Events

- [In.RoomEvent SavedEvent](#) = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.50.1 Event Documentation

5.50.1.1 In.RoomEvent Skylight.Save.SavedEvent = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/Save.cs

5.51 Skylight.Miscellaneous.SmileyIds.Shop Class Reference

Class [Shop](#).

Public Attributes

- const int [Ninja](#) = 12

The smilies that can be bought using the shop.

5.51.1 Detailed Description

Class [Shop](#).

5.51.2 Member Data Documentation

5.51.2.1 `const int Skylight.Miscellaneous.SmileyIds.Shop.Ninja = 12`

The smilies that can be bought using the shop.

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

- Skylight/Miscellaneous/SmileyIds.cs

5.52 Skylight.Show Class Reference

Public Member Functions

- **Show** ([In](#) @in)
- void **OnShow** ()

Events

- [In.RoomEvent ShowEvent](#) = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.52.1 Event Documentation

5.52.1.1 `In.RoomEvent Skylight.Show.ShowEvent = delegate { }`

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/Show.cs

5.53 Skylight.SignBlock Class Reference

Public Member Functions

- **SignBlock** ([In](#) @in)
- virtual void [OnSignBlockEvent](#) (Message m)
When a sign block is placed in the world.

Events

- [In.BlockEvent SignBlockEvent](#) = delegate { }

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

5.53.1 Member Function Documentation

5.53.1.1 `virtual void Skylight.SignBlock.OnSignBlockEvent (Message m) [virtual]`

When a sign block is placed in the world.

5.53.2 Event Documentation

5.53.2.1 `In.BlockEvent Skylight.SignBlock.SignBlockEvent = delegate { }`

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

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

- Skylight/SignBlock.cs

5.54 Skylight.Miscellaneous.SmileyIds Class Reference

Class [SmileyIds](#).

Classes

- class [Beta](#)
Class [Beta](#).
- class [Contest](#)
Class [Contest](#).
- class [Free](#)
Class [Free](#).
- class [Shop](#)
Class [Shop](#).
- class [Special](#)
Class [Special](#).
- class [Unimplemented](#)
Class [Unimplemented](#).

5.54.1 Detailed Description

Class [SmileyIds](#).

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

- Skylight/Miscellaneous/SmileyIds.cs

5.55 Skylight.Miscellaneous.SmileyIds.Special Class Reference

Class [Special](#).

Public Attributes

- const int [Superman](#) = 16
The special smilies that cannot be bought.

5.55.1 Detailed Description

Class [Special](#).

5.55.2 Member Data Documentation

5.55.2.1 `const int Skylight.Miscellaneous.SmileyIds.Special.Superman = 16`

The special smilies that cannot be bought.

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

- Skylight/Miscellaneous/SmileyIds.cs

5.56 Skylight.Tele Class Reference

Public Member Functions

- **Tele** ([In](#) @in)
- void **OnTele** (Message m)

Events

- [In.PlayerEvent DeathEvent](#) = delegate { }
*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*
- [In.RoomEvent ResetEvent](#) = delegate { }
Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.56.1 Event Documentation

5.56.1.1 `In.PlayerEvent Skylight.Tele.DeathEvent = delegate { }`

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

5.56.1.2 `In.RoomEvent Skylight.Tele.ResetEvent = delegate { }`

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/Tele.cs

5.57 Skylight.Teleport Class Reference

Public Member Functions

- **Teleport** ([In](#) @in)

- void **OnTeleport** (Message m)

Events

- [In.PlayerEvent TeleportEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.57.1 Event Documentation

5.57.1.1 [In.PlayerEvent Skylight.Teleport.TeleportEvent](#) = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

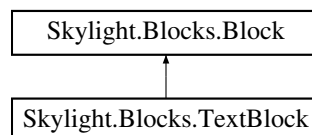
The documentation for this class was generated from the following file:

- Skylight/Teleport.cs

5.58 Skylight.Blocks.TextBlock Class Reference

Class [TextBlock](#).

Inheritance diagram for Skylight.Blocks.TextBlock:



Public Member Functions

- [TextBlock](#) (int id, int x, int y, string text)
Initializes a new instance of the [TextBlock](#) class.

Properties

- string [Text](#) [get, set]
Gets the text.

Additional Inherited Members

5.58.1 Detailed Description

Class [TextBlock](#).

5.58.2 Constructor & Destructor Documentation

5.58.2.1 Skylight.Blocks.TextBlock.TextBlock (int *id*, int *x*, int *y*, string *text*)

Initializes a new instance of the [TextBlock](#) class.

Parameters

<i>id</i>	The identifier.
<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.
<i>text</i>	The text.

5.58.3 Property Documentation

5.58.3.1 `string Skylight.Blocks.TextBlock.Text` `[get]`, `[set]`

Gets the text.

The text.

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

- Skylight/Blocks/TextBlock.cs

5.59 Skylight.Trophy Class Reference

Public Member Functions

- **Trophy** (`In @in`)
- void **OnTrophy** (Message m)

Events

- `In.PlayerEvent TrophyEvent` = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.59.1 Event Documentation

5.59.1.1 `In.PlayerEvent Skylight.Trophy.TrophyEvent` = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Trophy.cs

5.60 Skylight.Miscellaneous.SmileyIds.Unimplemented Class Reference

Class `Unimplemented`.

Public Attributes

- const int `Dj` = 50
The smilies that may or may not exist.

5.60.1 Detailed Description

Class [Unimplemented](#).

5.60.2 Member Data Documentation

5.60.2.1 `const int Skylight.Miscellaneous.SmileyIds.Unimplemented.Dj = 50`

The smilies that may or may not exist.

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

- Skylight/Miscellaneous/SmileyIds.cs

5.61 Skylight.Upgrade Class Reference

Public Member Functions

- **Upgrade** ([In](#) @in)
- void **OnUpgrade** ()

Events

- [In.RoomEvent](#) [UpdateEvent](#) = delegate { }

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

5.61.1 Event Documentation

5.61.1.1 `In.RoomEvent Skylight.Upgrade.UpdateEvent = delegate { }`

Delegates for RoomEvent. Are only invoked when commands that concern the room's state (such as global clear, potion toggling and saved) for just a few examples.

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

- Skylight/Upgrade.cs

5.62 Skylight.Witch Class Reference

Public Member Functions

- **Witch** ([In](#) @in)
- void **OnGiveWitch** (Message m)

Events

- [In.PlayerEvent](#) [WitchEvent](#) = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.62.1 Event Documentation

5.62.1.1 In.PlayerEvent Skylight.Witch.WitchEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Witch.cs

5.63 Skylight.Wizard Class Reference

Public Member Functions

- **Wizard** (In @in)
- void **OnGiveWizard** (Message m)

Events

- In.PlayerEvent WizardEvent = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.63.1 Event Documentation

5.63.1.1 In.PlayerEvent Skylight.Wizard.WizardEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/Wizard.cs

5.64 Skylight.WootUp Class Reference

Public Member Functions

- **WootUp** (In @in)
- void **OnWootUp** (Message m)

Events

- In.PlayerEvent WootEvent = delegate { }

*All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.*

5.64.1 Event Documentation

5.64.1.1 In.PlayerEvent Skylight.WootUp.WootEvent = delegate { }

All events that concern the player. This includes many messages that the player gets from the world (such as server information and leveling up). Mostly these events are shown from the server directly to the user in the form of a dialog box or by prefixing a chat message with *SYSTEM.

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

- Skylight/WootUp.cs

5.65 Skylight.Wp Class Reference

Public Member Functions

- **Wp** (In @in)
- void **OnWp** (Message m)

Events

- In.BlockEvent RoomPortalBlockEvent = delegate { }

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

5.65.1 Event Documentation

5.65.1.1 In.BlockEvent Skylight.Wp.RoomPortalBlockEvent = delegate { }

All of the delegates for BlockEvent. These fire when events occur (such as when a block was added or updated).

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

- Skylight/Wp.cs

5.66 Skylight.Write Class Reference

Public Member Functions

- **Write** (In @in)
- void **OnWrite** (Message m)

Events

- In.ChatEvent SystemMessageEvent = delegate { }

All of the delegates for ChatEvent. Chat events are when the player says something, and distinguishes between auto text and system messages and much more.

5.66.1 Event Documentation

5.66.1.1 In.ChatEvent Skylight.Write.SystemMessageEvent = delegate { }

All of the delegates for ChatEvent. [Chat](#) events are when the player says something, and distinguishes between auto text and system messages and much more.

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

- Skylight/Write.cs

Index

ArmorGames

Skylight::Bot, [18](#)

Facebook

Skylight::Bot, [18](#)

Kongregate

Skylight::Bot, [18](#)

Regular

Skylight::Bot, [18](#)

Skylight, [7](#)

Skylight::Bot

ArmorGames, [18](#)

Facebook, [18](#)

Kongregate, [18](#)

Regular, [18](#)