



Signalling Interface Parameters

Release version 1.0 (2007-11-20)

SIGNALLING INTERFACE PARAMETERS	1
1 ENGINE FUNCTION REFERENCE	3
1.1 ActivateNode	3
1.2 AddTime	3
1.3 Update	3
1.4 Reset	3
1.5 BeginUpdate	3
1.6 EndUpdate	3
1.7 GetSimulationTime	3
1.8 getNearPosition	4
1.9 setNearPosition	4
1.10 ControlExists	4
1.11 GetControlValue	4
1.12 SetControlValue	4
1.13 GetControlMinimum	4
1.14 GetControlMaximum	4
2 SIMULATION FUNCTION REFERENCE	5
2.1 BeginUpdate	5
2.2 EndUpdate	5
2.3 GetSimulationTime	5
3 SCRIPTING REFERENCE	6
3.1 Function names can be either:	6
3.2 Templates with a script component:	6
3.2.1 <i>cEngineBlueprint</i>	6
3.2.2 <i>cSignalBlueprint</i>	6
3.2.3 <i>cAnimSignalBlueprint</i>	6

4	SIGNALING FUNCTION REFERENCE	7
4.1	SendMessage	7
4.2	SendConsistMessage	7
4.3	GetNextSignalState	7
4.4	Set2DMapSignalState	7
4.5	SetSignalState(Deprecated)	8
4.6	GetConnectedLink	8
4.7	GetLinkCount	8
4.8	GetConsistSpeed	8
4.9	GetTrackSpeedLimit	8
4.10	GetConsistType	9
4.11	ActivateNode	9
4.12	AddTime	9
4.13	Reset	9
4.14	BeginUpdate	9
4.15	EndUpdate	9
4.16	GetSimulationTime	10
4.17	getNearPosition	10
4.18	setNearPosition	10

1 Engine Function Reference

1.1 ActivateNode

Description: Activate a node within the model.
Arguments: Node name - the name of the node to activate or "all"
Activate - whether to activate or deactivate the node
Returns: None
Usage: Used to turn on/off nodes in the object.

1.2 AddTime

Description: Move an animation forward/backward some amount of time
Arguments: Anim Id - The name of the animation to update
Time - the amount of time to move forward/backward in seconds +/- times allowed
Returns: Time - 0 if the animation has not reached its end or start otherwise the time remaining

1.3 Update

Usage: Used to update an animation

1.4 Reset

Description: Reset an animation to time 0
Arguments: Anim Id - The name of the animation to update
Returns: None
Usage: Used to reset an animation.

1.5 BeginUpdate

Description: Ask the game to start sending updates to this script.
Arguments: None
Returns: None
Usage: Used when a script needs to do updates over a number of frames.

1.6 EndUpdate

Description: Ask the game to stop sending updates to this script.
Arguments: None
Returns: None
Usage: Used when a script needs to do updates over a number of frames.

1.7 GetSimulationTime

Description: Get the time in seconds from the start of the session in game time.
Arguments: None
Returns: The time in seconds in the game session.

1.8 getNearPosition

Description: Get the x,y,z coordinate of the object in local frame coordinates
Arguments: None
Returns: x - the x co-ord
y - the y co-ord
z - the z co-ord

1.9 setNearPosition

Description: Get the x,y,z coordinate of the object in local frame coordinates
Arguments: x - the x co-ord
y - the y co-ord
z - the z cc-ord
Returns: None

1.10 ControlExists

Description: Find if the named control exists.
Arguments: Control name - the name of the control
Returns: True/False

1.11 GetControlValue

Description: Get the current value for a control
Arguments: Control name - the name of the control
Returns: Control value - the value of the control

1.12 SetControlValue

Description: Set the current value for a control
Arguments: Control name - the name of the control
Control value - the value of the control
Returns: None

1.13 GetControlMinimum

Description: Get the minimum value for a control
Arguments: Control name - the name of the control
Returns: Min value - the minimum value for the control

1.14 GetControlMaximum

Description: Get the maximum value for a control
Arguments: Control name - the name of the control
Returns: Max value - the maximum value for the control

2 Simulation Function Reference

2.1 BeginUpdate

Description: Ask the game to start sending updates to this script.
Arguments: None
Returns: None
Usage: Used when a script needs to do updates over a number of frames.

2.2 EndUpdate

Description: Ask the game to stop sending updates to this script.
Arguments: None
Returns: None
Usage: Used when a script needs to do updates over a number of frames.

2.3 GetSimulationTime

Description: Get the time in seconds from the start of the session in game time.
Arguments: None
Returns: The time in seconds in the game session.

3 Scripting Reference

3.1 Function names can be either:

- a) function
- b) *:function
- c) name:function

Function is the function to call; name is a name of an entity / child entity and * means entity and all child entities for (a) just the owning entity is searched for a matching function, (b) the owning entity and all descendants are searched, function is called the first component that implements the function
(c) A search is made of the owning entity and all descendants for the first called 'name' which implements function

3.2 Templates with a script component:

3.2.1 cEngineBlueprint

See Engine function reference (Only used in a very limited way currently)

3.2.2 cSignalBlueprint

See Signals function reference

3.2.3 cAnimSignalBlueprint

See Signals function reference

4 Signaling Function Reference

4.1 SendSignalMessage

Description: Send a message along the track to the next/previous signal link on the track (ignoring links of the same signal). See Signal message types for a description of messages.

Arguments: Message - The message type
Argument - An optional string argument passed with message
Look Direction - The direction the script sends the message relative to the link
1 = forwards, -1 = backwards
Link Direction - The direction of the link that the message should be sent to relative to the link
Link Index - The index of the link to send the message from

Returns: Signal Found - was a signal found
End Of Track - was the end of the track encountered (rather than a circuit)

Usage: Used to communicate between signals.

Special Cases: Where Argument is set to the special value "DoNotForward" the receiving signal is expected to not pass forward the message.

4.2 SendConsistMessage

Description: Send a message to the passing consists. See consist message types for a list of valid types

Arguments: Message - The message type
Argument - An optional string argument passed with message
Direction - Not used

Return: None

Usage: Only safe to use from OnConsistPassed. Used to indicate SPADS, AWS, TPWS etc

Special Cases: SIGMSG_CUSTOM - While the other message types are handled directly by the app, custom messages are passed on to the engine script using the script method OnCustomSignalMessage passing the Argument to the engine script.

4.3 GetNextSignalState

Description: Get the signal state of the next previous signal up/down the line.

Arguments: Argument - An optional string argument passed with the message (Not used)
Look Direction - The direction from the link 1 = forwards, -1 = backwards
Link Direction - The direction of the signal link 0 to look for
Link Index - The index of the link to start the search from

Return: The state of the next signal along the line where no signal is found Go is returned

Usage: Get the signal state of the next previous signal up/down the line.

4.4 Set2DMapSignalState

Description: Set the 2D map displayed state of the signal.

Arguments: State - The state to set from (go, warning, stop)
Usage: Set the 2D map displayed state of the signal.

4.5 SetSignalState(Deprecated)

Description: Set the code internal state of the signal. Not used anymore.
Arguments: State - The state to set from (go, warning, stop)
Usage: Deprecated

4.6 GetConnectedLink

Description: Get the index of the link currently connected by the track network to the link. If no link is connected then -1 is returned. Only finds links owned by the same signal.
Arguments: No longer used. (Previously used to limit search distance)
Direction - Not used. (Might be added in future to control bi-directional connections in a signal)
Link Index - The link to start at when looking for the next connected link. (Normally 0)
Usage: This is used as a method for testing which path ahead is set, usually in response to a junction change message or when initialising.
Special Cases: For slips and crossings (where the junction has 2 simultaneous legal paths) the dispatcher holds an internal state for the junction and as such a signal will get -1 as with a converging junction. In free roam and for slips set to the player's path the state is always set for the player's route.
Returns: the index of the next link within the same signal connected via the track network, if a converging junction set against the direction of the start link is found or the end of the track then -1 is returned.

4.7 GetLinkCount

Description: Get the number of links in the signal. - Blueprint::NumberOfTrackLinks
Arguments: None:
Returns: The number of links the signal has 1 - 10.

4.8 GetConsistSpeed

Description: Get the speed in m/s of the consist currently passing the signal link.
Arguments: None
Returns: The speed in m/s
Usage: Only safe to use from OnConsistPassed. Typically used for TPWS like systems.

4.9 GetTrackSpeedLimit

Description: Get the track speed limit at the link.
Arguments: Link Index - the index of the signal link
Returns: The speed in m/s
Usage: Used to test for speeding conditions in TPWS like systems.

4.10 GetConsistType

Description: Get the type of the consist currently passing the signal link.
Arguments: None
Returns: The type of train. (See train types)
Usage: Only safe to use from OnConsistPassed. Typically used for TPWS like systems.

4.11 ActivateNode

Description: Activate a node within the model.
Arguments: Node name - the name of the node to activate or "all"
Activate - whether to activate or deactivate the node
Returns: None
Usage: Used to turn on/off different aspects of a signal.

4.12 AddTime

Description: Move an animation forward/backward some amount of time
Arguments: Anim Id - The name of the animation to update
Time - the amount of time to move forward/backward in seconds +/- times allowed
Returns: Time - 0 if the animation has not reached its end or start otherwise the time remainder from the update
Usage: Used to animate arms/working parts within a signal

4.13 Reset

Description: Reset an animation to time 0
Arguments: Anim Id - The name of the animation to update
Returns: None
Usage: Used to reset an animation. Typically when a signal is reset after entering the editor

4.14 BeginUpdate

Description: Ask the game to start sending updates to this script.
Arguments: None
Returns: None
Usage: Used when a script needs to do updates over a number of frames.

4.15 EndUpdate

Description: Ask the game to stop sending updates to this script.
Arguments: None
Returns: None
Usage: Used when a script needs to do updates over a number of frames.

4.16 GetSimulationTime

Description: Get the time in seconds from the start of the session in game time.

Arguments: None

Returns: The time in seconds in the game session.

4.17 getNearPosition

Description: Get the x,y,z coordinate of the object in local frame coordinates

Arguments: None

Returns: x - the x co-ord

y - the y co-ord

z - the z co-ord

4.18 setNearPosition

Description: Get the x,y,z coordinate of the object in local frame coordinates

Arguments: x - the x co-ord

y - the y co-ord

z - the z co-ord

Returns: None