

1. HDK API Reference	3
1.1 HdkApi.Core.Collections	3
1.1.1 HdkApi.Core.Collections.HdkDictionary	3
1.2 HdkApi.Core.Communication	5
1.2.1 HdkApi.Core.Communication.Target	6
1.2.2 HdkApi.Core.Communication.TargetException	9
1.2.3 HdkApi.Core.Communication.TargetManager	9
1.2.4 HdkApi.Core.Communication.TargetResult	12
1.3 HdkApi.Core.Debug	12
1.3.1 HdkApi.Core.Debug.Logger	13
1.4 HdkApi.Core.Maths	14
1.4.1 HdkApi.Core.Maths.Converters	14
1.4.2 HdkApi.Core.Maths.Vector3	15
1.5 HdkApi.Core.Utilities	16
1.5.1 HdkApi.Core.Utilities.PathUtilities	17
1.6 HdkApi.Data.AgeRating	17
1.6.1 HdkApi.Data.AgeRating.AgeRatingItem	18
1.7 HdkApi.Data.Collect	24
1.7.1 HdkApi.Data.Collect.Collect	24
1.8 HdkApi.Data.Object	25
1.8.1 HdkApi.Data.Object.ComponentTypes	26
1.8.2 HdkApi.Data.Object.HomeObject	27
1.8.3 HdkApi.Data.Object.LocalisationItem	30
1.8.4 HdkApi.Data.Object.ObjectComponent	30
1.8.5 HdkApi.Data.Object.ObjectManager	33
1.8.6 HdkApi.Data.Object.ObjectResource	35
1.8.7 HdkApi.Data.Object.TerritoryTypes	36
1.8.8 HdkApi.Data.Object.Uuid	37
1.9 HdkApi.Data.Object.Components	38
1.9.1 HdkApi.Data.Object.Components.ClothingComponent	39
1.9.2 HdkApi.Data.Object.Components.FurnitureComponent	40
1.9.3 HdkApi.Data.Object.Components.GameLaunchComponent	41
1.9.4 HdkApi.Data.Object.Components.HeaderComponent	42
1.10 HdkApi.Data.Packager	45
1.10.1 HdkApi.Data.Packager.Packager	45
1.11 HdkApi.Data.Reporting	45
1.11.1 HdkApi.Data.Reporting.MessageType	46
1.12 HdkApi.Data.Scene	46
1.12.1 HdkApi.Data.Scene.GameObject	47
1.12.2 HdkApi.Data.Scene.GameObjectFolder	48
1.12.3 HdkApi.Data.Scene.GameObjectTypes	49
1.12.4 HdkApi.Data.Scene.HomeScene	50
1.12.5 HdkApi.Data.Scene.SceneConverter	51
1.12.6 HdkApi.Data.Scene.SceneTypes	52
1.13 HdkApi.Data.Scene.GameObjects	53
1.13.1 HdkApi.Data.Scene.GameObjects.GameObjectGroup	53
1.14 HdkApi.Data.Validator	54
1.14.1 HdkApi.Data.Validator.Validator	55
1.15 HdkApi.ObjectEditor	55
1.15.1 HdkApi.ObjectEditor.ObjectEditor	55
1.16 HdkApi.ObjectEditor.UserInterface	57
1.16.1 HdkApi.ObjectEditor.UserInterface.Menu	58
1.16.2 HdkApi.ObjectEditor.UserInterface.MenuItem	59
1.16.3 HdkApi.ObjectEditor.UserInterface.MenuManager	61
1.17 HdkApi.RepertoireEditor	62
1.17.1 HdkApi.RepertoireEditor.Action	65
1.17.2 HdkApi.RepertoireEditor.Behaviour	66
1.17.3 HdkApi.RepertoireEditor.BooleanRegister	69
1.17.4 HdkApi.RepertoireEditor.EaseCurve	69
1.17.5 HdkApi.RepertoireEditor.EaseCurveType	70
1.17.6 HdkApi.RepertoireEditor.IntegerRegister	71
1.17.7 HdkApi.RepertoireEditor.Locomotion	71
1.17.8 HdkApi.RepertoireEditor.LocomotionState	72
1.17.9 HdkApi.RepertoireEditor.NumberRegister	73
1.17.10 HdkApi.RepertoireEditor.Repertoire	73
1.17.11 HdkApi.RepertoireEditor.RepertoireEditor	75
1.17.12 HdkApi.RepertoireEditor.RepertoireEvent	77
1.17.13 HdkApi.RepertoireEditor.RepertoireRegister	78
1.17.14 HdkApi.RepertoireEditor.RepertoireTargetRig	78
1.17.15 HdkApi.RepertoireEditor.State	78
1.17.16 HdkApi.RepertoireEditor.StringRegister	79
1.17.17 HdkApi.RepertoireEditor.Transition	80
1.18 HdkApi.RepertoireEditor.Animation	81
1.18.1 HdkApi.RepertoireEditor.Animation.BlendNode	82
1.18.2 HdkApi.RepertoireEditor.Animation.BlendTree	83

1.18.3 HdkApi.RepertoireEditor.Animation.BlendTreeAnimation	84
1.18.4 HdkApi.RepertoireEditor.Animation.SkeletalAnimation	84
1.19 HdkApi.RepertoireEditor.Conditions	85
1.19.1 HdkApi.RepertoireEditor.Conditions.And	86
1.19.2 HdkApi.RepertoireEditor.Conditions.CompareCurrentBehaviour	87
1.19.3 HdkApi.RepertoireEditor.Conditions.CompareCurrentNetworkAge	88
1.19.4 HdkApi.RepertoireEditor.Conditions.CompareNetworkAge	88
1.19.5 HdkApi.RepertoireEditor.Conditions.ComparePreviousBehaviour	89
1.19.6 HdkApi.RepertoireEditor.Conditions.ComparePreviousState	89
1.19.7 HdkApi.RepertoireEditor.Conditions.CompareRegister	90
1.19.8 HdkApi.RepertoireEditor.Conditions.CompareStateAge	90
1.19.9 HdkApi.RepertoireEditor.Conditions.ComparisonType	91
1.19.10 HdkApi.RepertoireEditor.Conditions.Condition	91
1.19.11 HdkApi.RepertoireEditor.Conditions.Or	92
1.19.12 HdkApi.RepertoireEditor.Conditions.RandomValue	93
1.20 HdkApi.RepertoireEditor.Emotes	93
1.20.1 HdkApi.RepertoireEditor.Emotes.ActionEmote	94
1.20.2 HdkApi.RepertoireEditor.Emotes.BehaviourEmote	95
1.20.3 HdkApi.RepertoireEditor.Emotes.EmoteContext	97
1.20.4 HdkApi.RepertoireEditor.Emotes.EmoteGroup	98
1.20.5 HdkApi.RepertoireEditor.Emotes.GlobalEmotes	100
1.20.6 HdkApi.RepertoireEditor.Emotes.RegionalEmotes	101
1.21 HdkApi.RepertoireEditor.Outputs	103
1.21.1 HdkApi.RepertoireEditor.Outputs.ClearRegister	104
1.21.2 HdkApi.RepertoireEditor.Outputs.Output	105
1.21.3 HdkApi.RepertoireEditor.Outputs.OutputType	106
1.21.4 HdkApi.RepertoireEditor.Outputs.SetNextBehaviour	107
1.21.5 HdkApi.RepertoireEditor.Outputs.SetRegister	108
1.21.6 HdkApi.RepertoireEditor.Outputs.SpawnAction	109
1.21.7 HdkApi.RepertoireEditor.Outputs.TriggerEvent	110
1.22 HdkApi.RepertoireEditor.UserInterface	111
1.22.1 HdkApi.RepertoireEditor.UserInterface.Menu	112
1.22.2 HdkApi.RepertoireEditor.UserInterface.MenuItem	114
1.22.3 HdkApi.RepertoireEditor.UserInterface.MenuManager	117
1.23 HdkApi.SceneEditor	119
1.23.1 HdkApi.SceneEditor.SceneEditor	119
1.24 HdkApi.SceneEditor.Camera	122
1.24.1 HdkApi.SceneEditor.Camera.Camera	122
1.24.2 HdkApi.SceneEditor.Camera.CameraManager	123
1.24.3 HdkApi.SceneEditor.Camera.SceneCamera	124
1.25 HdkApi.SceneEditor.Selection	125
1.25.1 HdkApi.SceneEditor.Selection.SelectionManager	125
1.26 HdkApi.SceneEditor.UserInterface	126
1.26.1 HdkApi.SceneEditor.UserInterface.Menu	127
1.26.2 HdkApi.SceneEditor.UserInterface.MenuItem	128
1.26.3 HdkApi.SceneEditor.UserInterface.MenuManager	130
1.27 HdkApi.SceneEditor.Utilities	131
1.27.1 HdkApi.SceneEditor.Utilities.SynchronizationObject	132

HDK API Reference

HDK API is the full HDK scripting cookbook which allows you to add extra functionality to your tools using IronPython.

[Collapse all](#)

[Expand all](#) [Collapse all](#)

HdkApi.Core.Collections

Summary

Type
Description
Assembly
HdkDictionary
HdkDictionary is a light wrapper around a .NET Dictionary, it provides all the standard functionality of the .NET Dictionary.
Home.Core.Scripting

HdkApi.Core.Collections.HdkDictionary

HdkDictionary

HdkDictionary is a light wrapper around a .NET Dictionary, it provides all the standard functionality of the .NET Dictionary.

Namespace: [HdkApi.Core.Collections](#)

Assembly: *Home.Core.Scripting.dll*

Constructors

HdkDictionary()

HdkDictionary(resizable, dictionary)

HdkDictionary(equalityComparer)

HdkDictionary(size)

HdkDictionary(resizable, dictionary, equalityComparer)

HdkDictionary(size, equalityComparer)

Methods

Add(key, value)

System.Boolean ContainsKey(key)

System.Boolean Remove(key)

System.Boolean TryGetValue(key, value)

Add(item)

Clear()

System.Boolean Contains(item)

CopyTo(array, arrayIndex)

System.Boolean Remove(item)

System.Collections.Generic.IEnumerator[System.Collections.Generic.KeyValuePair[T1,T2]] GetEnumerator()

Properties

Keys

Values

[key]

Count

IsReadOnly

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# The HdkDictionary class must be imported from Collections before it is used
from HdkApi.Core.Collections import HdkDictionary

class HdkDictionaryExample():
    def __init__(self):

        # Create a new HdkDictionary mapping int to string
        dictionary = HdkDictionary[int, str]()

        # Add some new entries into the dictionary
        dictionary.Add(1, "red")
        dictionary.Add(2, "two")

        # Edit the value of an entry
        dictionary[1] = "one"

        # Entries can also be added by trying to edit a key which does not yet exist
        dictionary[3] = "three"
        dictionary[4] = "four"

        # Check that the dictionary contains a specific key
        if dictionary.ContainsKey(1):
            # Get the value associated with that key
            print dictionary[1]

            # Remove the entry with the key '1'
            dictionary.Remove(1)

        # Iterate through all the keys in the dictionary
        for key in dictionary.Keys:
            # Get each value
            print dictionary[key]

```

HdkApi.Core.Communication

Summary

Type
Description
Assembly
Target
Provides information about a PS3 target obtained from the TargetManager class. Once connected allows communications such as sending of messages.
Home.Core.Scripting

TargetException
Exception which provides a Result Code for a failed communications operation
Home.Core.Scripting
TargetManager
TargetManager is used to manage and connect to PS3 targets
Home.Core.Scripting
TargetResult
Result codes for operations with the Target Manager, can be accessed through the TargetException's ResultCode
Home.Core.Scripting

HdkApi.Core.Communication.Target

Target

Provides information about a PS3 target obtained from the TargetManager class. Once connected allows communications such as sending of messages.

Namespace: [HdkApi.Core.Communication](#)

Assembly: *Home.Core.Scripting.dll*

Methods

Connect()

ConnectAsync(connectCompleted)

Disconnect()

Send(command)

RegisterChannelOutput(channel)

UnregisterChannelOutput(channel)

Properties

Name

Ip

Port

Status

IsConnected

Events

ReceivedResponse

OnConnectionStatusChanged

Delegates

Response

ConnectCompleted

ConnectionStatus

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Communication classes
from HdKApi.Core.Communication import TargetManager, Target, TargetException, TargetResult

# Example of connecting to a target
class TargetManagerExample():
    def __init__(self):
        # Before using the TargetManager API you must call Initialize
        # Initialize automatically connects to the first Target it finds which already has a
        'connected' status
        # If there are no targets with a 'connected' status the TargetManager API will remain
        disconnected
        TargetManager.Initialize()

        # Check whether the TargetManager is connected to a target
        if TargetManager.ConnectedTarget == None:
            try:
                # Connect to the first available target
                # This function blocks until the connect is completed
                TargetManager.ConnectToDefault()
            except TargetException, e:
                # Catch any exceptions thrown by the connect
                print "TargetManagerExample failed to connect to default target:
{0}".format(e.ResultCode)

            return

        # Send a command enabling rendering of sound volumes in the client
        exampleCmd = "enableAudioRender3d 1"
        TargetManager.ConnectedTarget.Send(exampleCmd)

        # Send a lua command activating the editor camera in the client
        exampleLuaCmd = "lc Editor.ActivateEditorCamera()"
        TargetManager.ConnectedTarget.Send(exampleLuaCmd)

# Example of asynchronously connecting to a target
class TargetManagerAsyncExample():
    def __init__(self):
        # Before using the TargetManager API you must call Initialize
        TargetManager.Initialize()

        # Check whether the TargetManager is connected to a target
        if TargetManager.ConnectedTarget == None:
            try:
                # Connect to the first available target asynchronously
                # The ConnectCallback function will be called when the connect has completed
                TargetManager.ConnectToDefaultAsync(ConnectCallback)
            except TargetException, e:
                # Catch any exceptions thrown before the asynchronous connect begins
```

```
        # Any exceptions thrown will cause the connect to be aborted so the ConnectCallback
will never be called
        print "TargetManagerAsyncExample failed to connect to default target:
{0}".format(e.ResultCode)

        return

    else:
        # If the target is already connected proceed with sending the commands
        self.SendCommands()

def SendCommands(self):
    # Send a command enabling rendering of sound volumes in the client
    exampleCmd = "enableAudioRender3d 1"
    TargetManager.ConnectedTarget.Send(exampleCmd)

    # Send a lua command activating the editor camera in the client
    exampleLuaCmd = "lc Editor.ActivateEditorCamera()"
    TargetManager.ConnectedTarget.Send(exampleLuaCmd)

def ConnectCallback(self, target, result):
```

```
# Example of a callback function triggered by a completed asynchronous connect
if result == TargetResult.SUCCEEDED:
    self.SendCommands()
```

HdkApi.Core.Communication.TargetException

TargetException

Exception which provides a Result Code for a failed communications operation

Namespace: [HdkApi.Core.Communication](#)

Assembly: *Home.Core.Scripting.dll*

Properties

ResultCode

Message

HdkApi.Core.Communication.TargetManager

TargetManager

TargetManager is used to manage and connect to PS3 targets

Namespace: [HdkApi.Core.Communication](#)

Assembly: *Home.Core.Scripting.dll*

Remarks



Before attempting to connect to a target TargetManager must be initialised by calling TargetManager.Initialize()



If multiple targets are connected to through the SN Target Manager, only the first connected target will be connected to in the HdkApi TargetManager



TargetManager only supports connecting and communicating with one target at a time

Methods

TargetManager.Initialize()

System.Collections.Generic.Dictionary[System.String,HdkApi.Core.Communication.Target] TargetManager.ListTargets()

TargetManager.DisplayTargetsForm(connectCompleted)

TargetManager.ConnectToDefault()

TargetManager.ConnectToDefaultAsync(connectCompleted)

Properties

ConnectedTarget

Example

```
#<copyright>
# -----
#   SCE CONFIDENTIAL
```

```

# Home Development Kit (HDK) 1.55.0
# Copyright (C) 2011 Sony Computer Entertainment Inc.
# All Rights Reserved.
#-----
#</copyright>

# First import the Communication classes
from HdKApi.Core.Communication import TargetManager, Target, TargetException, TargetResult

# Example of connecting to a target
class TargetManagerExample():
    def __init__(self):
        # Before using the TargetManager API you must call Initialize
        # Initialize automatically connects to the first Target it finds which already has a
        'connected' status
        # If there are no targets with a 'connected' status the TargetManager API will remain
        disconnected
        TargetManager.Initialize()

        # Check whether the TargetManager is connected to a target
        if TargetManager.ConnectedTarget == None:
            try:
                # Connect to the first available target
                # This function blocks until the connect is completed
                TargetManager.ConnectToDefault()
            except TargetException, e:
                # Catch any exceptions thrown by the connect
                print "TargetManagerExample failed to connect to default target:
{0}".format(e.ResultCode)

            return

        # Send a command enabling rendering of sound volumes in the client
        exampleCmd = "enableAudioRender3d 1"
        TargetManager.ConnectedTarget.Send(exampleCmd)

        # Send a lua command activating the editor camera in the client
        exampleLuaCmd = "lc Editor.ActivateEditorCamera()"
        TargetManager.ConnectedTarget.Send(exampleLuaCmd)

# Example of asynchronously connecting to a target
class TargetManagerAsyncExample():
    def __init__(self):
        # Before using the TargetManager API you must call Initialize
        TargetManager.Initialize()

        # Check whether the TargetManager is connected to a target
        if TargetManager.ConnectedTarget == None:
            try:
                # Connect to the first available target asynchronously
                # The ConnectCallback function will be called when the connect has completed
                TargetManager.ConnectToDefaultAsync(ConnectCallback)
            except TargetException, e:
                # Catch any exceptions thrown before the asynchronous connect begins
                # Any exceptions thrown will cause the connect to be aborted so the ConnectCallback
                will never be called
                print "TargetManagerAsyncExample failed to connect to default target:
{0}".format(e.ResultCode)

            return
        else:
            # If the target is already connected proceed with sending the commands
            self.SendCommands()

    def SendCommands(self):
        # Send a command enabling rendering of sound volumes in the client
        exampleCmd = "enableAudioRender3d 1"
        TargetManager.ConnectedTarget.Send(exampleCmd)

        # Send a lua command activating the editor camera in the client

```

```
exampleLuaCmd = "lc Editor.ActivateEditorCamera()"
TargetManager.ConnectedTarget.Send(exampleLuaCmd)
```

```
def ConnectCallback(self, target, result):
```

```
# Example of a callback function triggered by a completed asynchronous connect
if result == TargetResult.SUCCEEDED:
    self.SendCommands()
```

HdkApi.Core.Communication.TargetResult

TargetResult

Result codes for operations with the Target Manager, can be accessed through the TargetException's ResultCode

Namespace: [HdkApi.Core.Communication](#)

Assembly: *Home.Core.Scripting.dll*

Fields

SUCCEEDED

TM_FAILED_TO_START

TARGET_COMMS_ERROR

TARGET_IN_USE

TM_NO_TARGETS

TARGET_NOT_FOUND

TARGET_NOT_CONNECTED

TM_NOT_INITIALIZED

FAILED

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Communication classes
from HdkApi.Core.Communication import TargetManager, Target, TargetException, TargetResult

# Example of using Target Result
class TargetResultExample():
    def __init__(self):
        try:
            TargetManager.Initialise()
        except TargetException, e:
            # Print the Target Result for the exception
            print e.ResultCode

            # Check for a specific Target Result
            if e.ResultCode == TargetResult.TM_FAILED_TO_START:
                print "Target Manager failed to start"
                # Run some fallback code
```

HdkApi.Core.Debug

Summary

Type
Description
Assembly
Logger
Provides simple file logging functionality, allowing output from a program to be written to a log file.
Home.Core.Scripting

HdkApi.Core.Debug.Logger

Logger

Provides simple file logging functionality, allowing output from a program to be written to a log file.

Namespace: [HdkApi.Core.Debug](#)
Assembly: *Home.Core.Scripting.dll*

Constructors

Logger(logFilePath)

Methods

ClearLog()

LogInfo(info)

LogWarning(warning)

LogError(error)

LogFatal(fatal)

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# The Logger class must be imported from Debug before it is used
from HdkApi.Core.Debug import Logger

class LoggerExample():
    def __init__(self):

        # Create a new logger which writes to the file C:/MyLog.txt
        logger = Logger("C:/MyLog.txt")

        # Write an information message to the log
        logger.LogInfo("Example started")

        # Write a warning message to the log
        logger.LogWarning("Example warning")

        # Write an error message to the log
        logger.LogError(("Example error"))

        try:
            raise Exception("Example fatal error")
        except Exception as e:
            # Write a fatal error message to the log
            logger.LogFatal(str(e))

```

HdkApi.Core.Maths

Summary

Type

Description

Assembly

Converters

Provides helper functions for performing various mathematical conversions such as converting radian values to degrees.

Home.Core.Scripting

Vector3

A simple vector structure containing 3 float components.

Home.Core.Scripting

HdkApi.Core.Maths.Converters

Converters

Provides helper functions for performing various mathematical conversions such as converting radian values to degrees.

Namespace: [HdkApi.Core.Maths](#)

Assembly: *Home.Core.Scripting.dll*

Methods

System.Double Converters.RadToDeg(radians)

System.Double Converters.DegToRad(degrees)

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# The Converters class must be imported from Maths before it is used
from HdkApi.Core.Maths import Converters

class ConvertersExample():
    def __init__(self):

        angleInDegrees = 180

        # Convert our angle in degrees to radians
        angleInRadians = Converters.DegToRad(angleInDegrees)
        print angleInRadians

        # Convert our angle in radians to degrees
        angleInDegrees = Converters.RadToDeg(angleInRadians)
        print angleInDegrees
```

HdkApi.Core.Maths.Vector3

Vector3

A simple vector structure containing 3 float components.

Namespace: [HdkApi.Core.Maths](#)

Assembly: *Home.Core.Scripting.dll*

Constructors

Vector3(x, y, z)

Methods

System.String ToString(format, formatProvider)

Properties

X
Y
Z

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Vector3 class from Maths
from HdkApi.Core.Maths import Vector3

class Vector3Example():
    def __init__(self):
        # Example of creating Vector3 objects
        vecA = Vector3(1.0, 1.0, 1.0)
        vecB = Vector3(1.0, 1.0, 1.0)

        # Various arithmetic operations can be performed on the Vector3 objects
        vecC = vecA + vecB
        vecD = vecA - vecB
        vecE = vecA * vecB
        vecF = vecA * 5
        vecG = vecA / vecB
        vecH = vecA / 2

        # Comparison operations can also be performed
        if vecA == vecB:
            print "Vectors are equal"

        if vecA != vecB:
            print "Vectors are not equal"

        # The X, Y and Z components of a Vector3 can be accessed through properties
        print vecC.X
        print vecC.Y
        print vecC.Z
```

HdkApi.Core.Utilities

Summary

Type
Description
Assembly
PathUtilities
PathUtilities can be used to get the current Project path and the HDK install root. These paths are useful for referring to scripts or other files relative from your project or HDK install.

HdkApi.Core.Utilities.PathUtilities

PathUtilities

PathUtilities can be used to get the current Project path and the HDK install root. These paths are useful for referring to scripts or other files relative from your project or HDK install.

Namespace: [HdkApi.Core.Utilities](#)

Assembly: *Home.Core.Scripting.dll*

Properties

ProjectPath

HdkInstallRoot

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# The PathUtilities class must be imported from Utilities before it is used
from HdkApi.Core.Utilities import PathUtilities

class PathUtilitiesExample():
    def __init__(self):

        # Get the current project path
        projectPath = PathUtilities.ProjectPath

        filePath = System.IO.Path.Combine(projectPath, "myFile.txt")
        print(filePath)

        # Get the current hdk install root
        hdkInstallRoot = PathUtilities.HdkInstallRoot

        iconPath = System.IO.Path.Combine(hdkInstallRoot, "dev/scripts/SceneEditor/icons/myFile.png")
        print(iconPath)
```

HdkApi.Data.AgeRating

Summary

Type
Description
Assembly

AgeRatingItem
Provides information about an age rating item
Home.Data.Scripting

HdkApi.Data.AgeRating.AgeRatingItem

AgeRatingItem

Provides information about an age rating item

Namespace: [HdkApi.Data.AgeRating](#)
Assembly: *Home.Data.Scripting.dll*

Additional Information

Country	Code	Region	Rating System
United States	US	SCEA	ESRB
Canada	CA	SCEA	ESRB
United Kingdom	GB	SCEE	PEGI BBFC
Ireland	IE	SCEE	PEGI
Belgium	BE	SCEE	PEGI
Luxembourg	LU	SCEE	PEGI
Netherlands	NL	SCEE	PEGI
France	FR	SCEE	PEGI
Germany	DE	SCEE	USK
Austria	AT	SCEE	PEGI
Switzerland	CH	SCEE	PEGI
Italy	IT	SCEE	PEGI
Portugal	PT	SCEE	PEGI
Denmark	DK	SCEE	PEGI
Finland	FI	SCEE	PEGI
Norway	NO	SCEE	PEGI
Sweden	SE	SCEE	PEGI
Australia	AU	SCEE	ACB
New Zealand	NZ	SCEE	OFLCNZ ACB
Spain	ES	SCEE	PEGI
Russia	RU	SCEE	PEGI
UAE	AE	SCEE	PEGI
South Africa	ZA	SCEE	PEGI
Czech Republic	CZ	SCEE	PEGI
Saudi Arabia	SA	SCEE	PEGI

Poland	PL	SCEE	PEGI
Greece	GR	SCEE	PEGI
Hong Kong	HK	SCEAsia	
Taiwan	TW	SCEAsia	CGSRR
Singapore	SG	SCEAsia	
Malaysia	MY	SCEAsia	
Thailand	TH	SCEAsia	
Indonesia	ID	SCEAsia	
Korea	KR	SCEK	GRB
Japan	JP	SCEJ	SCEJ

Rating System	Rating Code	Description	Descriptor
ESRB	EC E E10+ T M AO RP	Early Childhood Everyone Everyone 10+ Teen Mature Adults Only Rating Pending	Alcohol Reference Animated Blood Blood Blood and Gore Cartoon Violence Comic Mischief Crude Humour Drug Reference Edutainment Fantasy Violence Informational Intense Violence Language Lyrics Mature Humour Mild Blood Mild Cartoon Violence Mild Fantasy Violence Mild Language Mild Lyrics Mild Violence Mild Suggestive Themes Nudity Partial Nudity Real Gambling Sexual Themes Sexual Violence Simulated Gambling Some Assistance May Be Needed Strong Language Strong Lyrics Strong Sexual Content Suggestive Themes Tobacco References Use of Alcohol Use of Drugs Use of Tobacco Violence Online Interactions Not Rated by the ESRB Visit www.esrb.org for rating information. May contain content inappropriate for children.
PEGI	OK 3+ 4+ 6+ 7+ 12+ 16+ 18+	PEGI OK PEGI 3+ PEGI 4+ PEGI 6+ PEGI 7+ PEGI 12+ PEGI 16+ PEGI 18+	Bad Language Discrimination Drugs Fear Sexual Content Violence Gambling Online

BBFC	Uc U PG 12A 15 18 R18	Universal Children Universal Parental Guidance 12 Advisory 15 18 Restricted 18	
USK	0 6 12 16 18	0 6 12 16 18	
ACB	CTC G PG M MA15+	Check The Classification General Parental Guidance Mature Mature Restricted	Blood and Gore Content May Change When Played Online Crude Humor Dangerous Stunts Mild Battle Violence Mild Coarse Language Mild Drug References Mild Fantasy Violence Mild Gambling References Mild Horror Theme Mild Nudity Mild Science Fiction Violence Mild Sex Scene Mild Sex Scenes Mild Sexual Reference Mild Sexual References Mild Themes Mild Violence Moderate Battle Violence Moderate Coarse Language Moderate Drug References Moderate Drug Use Moderate Fantasy Violence Moderate Horror Themes Moderate Horror Violence Moderate Nudity Moderate Science Fiction Violence Moderate Sex Scene Moderate Sex Scenes Moderate Sexual Reference Moderate Sexual References Moderate Themes Moderate Violence Scary Scenes Strong Battle Violence Strong Coarse Language Strong Drug References Strong Drug Use Strong Fantasy Violence Strong Graphic Violence Strong Horror Themes Strong Horror Violence Strong Nudity Strong Science Fiction Violence Strong Sex Scene Strong Sex Scenes Strong Sexual Reference Strong Sexual References Strong Sexual Violence Strong Themes Strong Violence

OFLCNZ	G PG M R13 R15 R16 R18 R	General Parental Guidance Mature Restricted 13 Restricted 15 Restricted 16 Restricted 18 Restricted	Adult Themes Anti-Social Behaviour Battle Violence Brutal Violence Content That May Disturb Content That May Offend Could Have a Negative Effect on Children Cruelty Depicts Graphic And Realistic War Scenes Disturbing Content Drug Use Explicit Language Explicit Sex Scenes Extra Material May Offend or Disturb Graphic Animal Cruelty Graphic Violence Horror Scenes Irresponsible Behaviour Associated With Alcohol Nudity Offensive Language Other Content That May Offend Realistic Horror Sadistic Cruelty Sadistic Violence Sex Scenes Sexual Abuse Sexual Abuse Themes Sexual Coercion Sexual Content That May Offend Sexual Material Sexual References Sexual Themes Sexual Violence Strong Offensive Language Suicide Torture Use of Menstrual Blood Violence
GRB	ALL 12 15 18	ALL 12 15 18	Violence Sexual Horror Language Drug Crime Gamble
SCEJ	B C D Z	12+ 15+ 17+ 18+	Love Sexual Content Violence Horror Gambling Crime Use of alcohol or tobacco Use of drugs Language
CGSRR	All Ages 6 and above 12 and above 18 and above	All Ages 6 and above 12 and above 18 and above	

Methods

HdkApi.Data.AgeRating.AgeRatingItem AgeRatingItem.Create(rating)

Properties

Rating

MinimumAge

ParentalControlLevel

ContentDescriptors

AdditionalData

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# Import the Object classes
from HdkApi.Data.Object import ObjectManager, Uuid, ComponentTypes

# Import the AgeRatingItem class
from HdkApi.Data.AgeRating import AgeRatingItem

class AgeRatingsExample():
    def __init__(self):

        uuid = Uuid.Create("00000000-00000000-00000000-00000018")
        loadedObject = None

        try:
            # First load the object whose age rating data we wish to edit
            loadedObject = ObjectManager.LoadByUuid(uuid)

        except Exception, e:
            print "Failed to load object '{0}': {1}".format(uuid, e)
            return

        if not loadedObject:
            print "Object '{0}' was not found".format(uuid)
            return

        # An object's age ratings are accessed through its header component
        header = loadedObject.Components[ComponentTypes.Header]

        # The age ratings are returned as a HdkDictionary keyed by the country which each rating
        belongs to
        for ageRating in header.AgeRatings:
            print "{0} - {1}".format(ageRating.Key, ageRating.Value.Rating)

        # To create a new age rating use the AgeRatingItem.Create method
        # This method takes in a string containing the rating system and rating
        pegiItem = AgeRatingItem.Create("PEGI_12+")

        # To set the Parental Control Level use the ParentalControlLevel property
        pegiItem.ParentalControlLevel = 8

        # To set the Minimum Age use the MinimumAge property
        pegiItem.MinimumAge = 12

        # To set a particular content descriptor use the ContentDescriptors property
        # This property returns a HdkDictionary of descriptors to booleans
        # By default all descriptors are set to False
        pegiItem.ContentDescriptors["Violence"] = True

        # Once the age rating has been setup it can be assigned to a country
        # If the country does not support the age rating's rating system an exception is thrown
```

```
header.AgeRatings["GB"] = pegiItem

# The same age rating can be assigned to multiple countries
header.AgeRatings["FR"] = pegiItem

# An age rating can also be assigned to all countries which use a particular rating system
header.AgeRatings["ALL-PEGI"] = pegiItem

# To remove an age rating use the Remove method of the HdkDictionary
header.AgeRatings.Remove("ALL-PEGI")
```

```
# To write the changed age ratings to disc use the HomeObject Save method  
loadedObject.Save()
```

HdkApi.Data.Collect

Summary

Type
Description
Assembly
Collect
A utility class used to collect dependencies of objects or scenes.
Home.Data.Scripting

HdkApi.Data.Collect.Collect

Collect

A utility class used to collect dependencies of objects or scenes.

Namespace: [HdkApi.Data.Collect](#)

Assembly: *Home.Data.Scripting.dll*

Methods

System.String[] Collect.DependenciesForObject(obj)

System.String[] Collect.DependenciesForObjects(objs)

System.String[] Collect.DependenciesForScene(scene)

System.String[] Collect.DependenciesForScenes(scenes)

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
from HdkApi.Data.Object import ObjectManager
from HdkApi.Data.Collect import Collect
import zipfile

class CollectExample():
    def __init__(self):
        components = ["network"]
        foundObjects = ObjectManager.SearchByComponents(components)
        dependentFiles = Collect.DependenciesForObjects(foundObjects)

        f = zipfile.ZipFile('d:\\collect_test.zip', 'w')
        for dependentFile in dependentFiles:
            f.write(dependentFile)
        f.close()

collectExample = CollectExample()

```

Associated Types

- [HdkApi.Data.Object.Uuid](#)

HdkApi.Data.Object

Summary

Type
Description
Assembly
ObjectComponent
Provides information about an object component
Home.Data.Scripting
ComponentTypes
Lists all of the object component types
Home.Data.Scripting
HomeObject
Provides information about an object loaded through the ObjectManager class. Allows an object's data to be viewed and edited.

Home.Data.Scripting
LocalisationItem
Home.Data.Scripting
ObjectManager
ObjectManager is used to load objects for viewing and editing
Home.Data.Scripting
ObjectResource
Provides information about a resource on disk which may be associated with an object and used by a component
Home.Data.Scripting
TerritoryTypes
Lists all of the territory types
Home.Data.Scripting
Uuid
Uuid object used to identify an object. To create a new Uuid use Uuid.Create()
Home.Data.Scripting

HdkApi.Data.Object.ComponentTypes

ComponentTypes

Lists all of the object component types

Namespace: [HdkApi.Data.Object](#)
Assembly: *Home.Data.Scripting.dll*

Remarks

 Each type is used to look up a specific component in the Components dictionary accessed through an Object's Components property

Constructors

ComponentTypes()

Fields

Active

ArcadeGame
Clothing
EmbeddedObject
Furniture
GameLaunch
Header
MiniGame
Portable
Targetable
LuaEnvironment
Entity
Particles
Network
EventTimer
Osd
Pad
Renderer
Screen
Camera
System
GameLaunchProperties
Light
Repertoire
GameSpawner
RealTimeGame
GroupDoor
ResourcePack
MemorySlots
SceneTransition

Associated Types

- [HdkApi.Data.Object.HomeObject](#)
- [HdkApi.Data.Object.ObjectComponent](#)

HdkApi.Data.Object.HomeObject

HomeObject

Provides information about an object loaded through the `ObjectManager` class. Allows an object's data to be viewed and edited.

Namespace: [HdkApi.Data.Object](#)

Assembly: *Home.Data.Scripting.dll*

Methods

System.Boolean Validate(validatorOutput)

System.Boolean Package(validatorOutput, packagerOutput)

Save()

System.Collections.Generic.List[HdkApi.Data.Object.ObjectResource] GetResources()

System.Collections.Generic.List[HdkApi.Data.Object.ObjectResource] GetResources(type)

HdkApi.Data.Object.ObjectResource AddResource(name, type, filepath)

RemoveResource(resource)

System.Collections.Generic.List[HdkApi.Data.Object.LocalisationItem] GetLocalisationItems()

HdkApi.Data.Object.LocalisationItem GetLocalisationItem(key)

HdkApi.Data.Object.LocalisationItem AddLocalisationItem(reference)

RemoveLocalisationItem(item)

Properties

Uuid

Name

Components

EntitlementId

CategoryId

ProductId

Dependencies

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Object classes
from HdkApi.Data.Object import ObjectManager, HomeObject, ObjectComponent, Uuid, ComponentTypes,
TerritoryTypes

# Import the MessageType class as we need this for the object validation and packaging output
from HdkApi.Data.Reporting import MessageType

# Also import PathUtilities as we use it to get the project path
from HdkApi.Core.Utilities import PathUtilities

class ObjectManagerExample():
    def __init__(self):
        try:
            # Load an object by its uuid
            uuid = Uuid.Create("00000000-00000000-00000000-00000018")
            loadedObject = ObjectManager.LoadByUuid(uuid)
```

```

        # Load an object by its object.xml path
        objectPath = PathUtilities.ProjectPath +
"/build/objects/00000000-00000000-00000000-00000018/object.xml"
        loadedObject = ObjectManager.LoadByFilename(objectPath)

    except Exception, e:
        print "Failed to load object {0}: {1}".format(uuid, e)

    # Get the Uuids of all objects which contain either a 'furniture' or an 'active' component
    components = ["furniture", "active"]
    foundObjects = ObjectManager.SearchByComponents(components)

    # Get the Uuids of all objects which contain a resource with the path 'resourcePath'
    resourcePath = "objects/00000000-00000000-00000000-00000018/main.lua"
    foundObjects = ObjectManager.SearchByResource(resourcePath)

    # Print the entitlement_id for the territory 'SCEA'
    print loadedObject.EntitlementId[TerritoryTypes.SCEA]

    # Set the entitlement_id for the territory 'SCEE'
    loadedObject.EntitlementId[TerritoryTypes.SCEE] = "LUA_REWARD"

    # Print the name of each component which loadedObject contains
    for component in loadedObject.Components.Keys:
        print component

    # Try to access the furniture component and set the furniture metadata item
    if loadedObject.Components.ContainsKey(ComponentTypes.Furniture):
        loadedObject.Components[ComponentTypes.Furniture].Furniture = "Chair"

    # Validate an object
    if loadedObject.Validate(self.Output):
        print "{0} - {1} passed validations\n".format(loadedObject.Name, loadedObject.Uuid)

    # Validate and package an object
    # The first passed in callback is triggered for validation output
    # The second callback is triggered for packager output
    if loadedObject.Package(self.Output, self.Output):
        print "{0} - {1} succesfully packaged\n".format(loadedObject.Name, loadedObject.Uuid)

# Example function for handling object validation and packager output
def Output(self, msg, msgType):
    if (msgType == MessageType.INFO):
        print "INFO: {0}\n".format(msg)

    elif (msgType == MessageType.WARNING):
        print "WARNING: {0}\n".format(msg)

    elif (msgType == MessageType.ERROR):
        print "ERROR: {0}\n".format(msg)

    elif (msgType == MessageType.FATAL):

```

```
print "FATAL: {0}\n".format(msg)

objectManagerExample = ObjectManagerExample()
```

Associated Types

- [HdkApi.Data.Object.ObjectManager](#)
- [HdkApi.Data.Object.ObjectComponent](#)

HdkApi.Data.Object.LocalisationItem

LocalisationItem

Namespace: [HdkApi.Data.Object](#)
Assembly: *Home.Data.Scripting.dll*

Methods

System.Boolean HasLanguage(language)
System.String GetPhrase(language)
SetPhrase(language, phrase)
HdkApi.Core.Collections.HdkDictionary[System.String,System.String] GetPhrases()

Properties

Key
SupportedLanguages

Associated Types

- [HdkApi.Data.Object.ObjectManager](#)
- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.ObjectComponent

ObjectComponent

Provides information about an object component

Namespace: [HdkApi.Data.Object](#)
Assembly: *Home.Data.Scripting.dll*

Properties

Type

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Object classes
from HdkApi.Data.Object import ObjectManager, HomeObject, ObjectComponent, Uuid, ComponentTypes,
TerritoryTypes

# Import the MessageType class as we need this for the object validation and packaging output
from HdkApi.Data.Reporting import MessageType

# Also import PathUtilities as we use it to get the project path
from HdkApi.Core.Utilities import PathUtilities

class ObjectManagerExample():
    def __init__(self):
        try:
            # Load an object by its uuid
            uuid = Uuid.Create("00000000-00000000-00000000-00000018")
            loadedObject = ObjectManager.LoadByUuid(uuid)

            # Load an object by its object.xml path
            objectPath = PathUtilities.ProjectPath +
"/build/objects/00000000-00000000-00000000-00000018/object.xml"
            loadedObject = ObjectManager.LoadByFilename(objectPath)

        except Exception, e:
            print "Failed to load object {0}: {1}".format(uuid, e)

        # Get the Uuids of all objects which contain either a 'furniture' or an 'active' component
        components = ["furniture", "active"]
        foundObjects = ObjectManager.SearchByComponents(components)

        # Get the Uuids of all objects which contain a resource with the path 'resourcePath'
        resourcePath = "objects/00000000-00000000-00000000-00000018/main.lua"
        foundObjects = ObjectManager.SearchByResource(resourcePath)

        # Print the entitlement_id for the territory 'SCEA'
        print loadedObject.EntitlementId[TerritoryTypes.SCEA]

        # Set the entitlement_id for the territory 'SCEE'
        loadedObject.EntitlementId[TerritoryTypes.SCEE] = "LUA_REWARD"

        # Print the name of each component which loadedObject contains
        for component in loadedObject.Components.Keys:
            print component

        # Try to access the furniture component and set the furniture metadata item
        if loadedObject.Components.ContainsKey(ComponentTypes.Furniture):
            loadedObject.Components[ComponentTypes.Furniture].Furniture = "Chair"

        # Validate an object
        if loadedObject.Validate(self.Output):
            print "{0} - {1} passed validations\n".format(loadedObject.Name, loadedObject.Uuid)

        # Validate and package an object
        # The first passed in callback is triggered for validation output
        # The second callback is triggered for packager output
        if loadedObject.Package(self.Output, self.Output):
            print "{0} - {1} succesfully packaged\n".format(loadedObject.Name, loadedObject.Uuid)

        # Example function for handling object validation and packager output
        def Output(self, msg, msgType):
            if (msgType == MessageType.INFO):
                print "INFO: {0}\n".format(msg)

```

```
elif (msgType == MessageType.WARNING):  
    print "WARNING: {0}\n".format(msg)  
  
elif (msgType == MessageType.ERROR):  
    print "ERROR: {0}\n".format(msg)  
  
elif (msgType == MessageType.FATAL):
```

```
print "FATAL: {0}\n".format(msg)

objectManagerExample = ObjectManagerExample()
```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.ObjectManager

ObjectManager

ObjectManager is used to load objects for viewing and editing

Namespace: [HdkApi.Data.Object](#)

Assembly: *Home.Data.Scripting.dll*

Methods

HdkApi.Data.Object.HomeObject ObjectManager.LoadByFilename(objectFilename)

HdkApi.Data.Object.HomeObject ObjectManager.LoadByUuid(uuidToLoad)

System.Collections.Generic.IEnumerable[HdkApi.Data.Object.Uuid] ObjectManager.SearchByComponents(components)

System.Collections.Generic.IEnumerable[HdkApi.Data.Object.Uuid] ObjectManager.SearchByResource(resourcePath)

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Object classes
from HdkApi.Data.Object import ObjectManager, HomeObject, ObjectComponent, Uuid, ComponentTypes,
TerritoryTypes

# Import the MessageType class as we need this for the object validation and packaging output
from HdkApi.Data.Reporting import MessageType

# Also import PathUtilities as we use it to get the project path
from HdkApi.Core.Utilities import PathUtilities

class ObjectManagerExample():
    def __init__(self):
        try:
            # Load an object by its uuid
            uuid = Uuid.Create("00000000-00000000-00000000-00000018")
            loadedObject = ObjectManager.LoadByUuid(uuid)

            # Load an object by its object.xml path
            objectPath = PathUtilities.ProjectPath +
"/build/objects/00000000-00000000-00000000-00000018/object.xml"
            loadedObject = ObjectManager.LoadByFilename(objectPath)

        except Exception, e:
```

```

        print "Failed to load object {0}: {1}".format(uuid, e)

# Get the Uuids of all objects which contain either a 'furniture' or an 'active' component
components = ["furniture", "active"]
foundObjects = ObjectManager.SearchByComponents(components)

# Get the Uuids of all objects which contain a resource with the path 'resourcePath'
resourcePath = "objects/00000000-00000000-00000000-00000018/main.lua"
foundObjects = ObjectManager.SearchByResource(resourcePath)

# Print the entitlement_id for the territory 'SCEA'
print loadedObject.EntitlementId[TerritoryTypes.SCEA]

# Set the entitlement_id for the territory 'SCEE'
loadedObject.EntitlementId[TerritoryTypes.SCEE] = "LUA_REWARD"

# Print the name of each component which loadedObject contains
for component in loadedObject.Components.Keys:
    print component

# Try to access the furniture component and set the furniture metadata item
if loadedObject.Components.ContainsKey(ComponentTypes.Furniture):
    loadedObject.Components[ComponentTypes.Furniture].Furniture = "Chair"

# Validate an object
if loadedObject.Validate(self.Output):
    print "{0} - {1} passed validations\n".format(loadedObject.Name, loadedObject.Uuid)

# Validate and package an object
# The first passed in callback is triggered for validation output
# The second callback is triggered for packager output
if loadedObject.Package(self.Output, self.Output):
    print "{0} - {1} succesfully packaged\n".format(loadedObject.Name, loadedObject.Uuid)

# Example function for handling object validation and packager output
def Output(self, msg, msgType):
    if (msgType == MessageType.INFO):
        print "INFO: {0}\n".format(msg)

    elif (msgType == MessageType.WARNING):
        print "WARNING: {0}\n".format(msg)

    elif (msgType == MessageType.ERROR):
        print "ERROR: {0}\n".format(msg)

    elif (msgType == MessageType.FATAL):

```

```
print "FATAL: {0}\n".format(msg)

objectManagerExample = ObjectManagerExample()
```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.ObjectResource

ObjectResource

Provides information about a resource on disk which may be associated with an object and used by a component

Namespace: [HdkApi.Data.Object](#)

Assembly: *Home.Data.Scripting.dll*

Additional Information

Resource Types:

- "model"
- "texture"
- "particle"
- "file"
- "xml"
- "collision"
- "anim"
- "skeleton"
- "soundbank"
- "soundstream"
- "repertoire"
- "rig_settings"
- "control_rig"
- "emotes"
- "lua"
- "image"

Properties

Path

Type

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Object classes
from HdkApi.Data.Object import ObjectManager, HomeObject, Uuid, ObjectResource

# Also import PathUtilities as we use it to get the project path
from HdkApi.Core.Utilities import PathUtilities

class HomeObjectResourceExample():
    def __init__(self):
        try:
            # Load an object by its uuid
            uuid = Uuid.Create("00000000-00000000-00000000-00000018")
            loadedObject = ObjectManager.LoadByUuid(uuid)

            # Add model resource to the object
            resourcePath = PathUtilities.ProjectPath +
'/Build/Furniture/Home_Rug_Sample/Home_Rug_Sample.mdl'
            resource = loadedObject.AddResource("test_resource", 'model', resourcePath)

            # Print the name of each resource the loadedObject contains
            for objectResource in loadedObject.GetResources():
                print objectResource.Name

            # Print the name of each model resource the loadedObject contains
            for objectResource in loadedObject.GetResources('model'):
                print objectResource.Name

            # Remove the resource from the object
            loadedObject.RemoveResource(resource)

        except Exception, e:
            print "{0}".format(e)

resourceExample = HomeObjectResourceExample()

```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.TerritoryTypes

TerritoryTypes

Lists all of the territory types

Namespace: [HdkApi.Data.Object](#)

Assembly: *Home.Data.Scripting.dll*

Remarks



Each type is used to look up a specific value in the entitlement_id, category_id and product_id dictionaries accessed through an Object's EntitlementId, CategoryId and ProductId properties

Fields

SCEE

SCEA

SCEJ

SCEAsia

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the TerritoryTypes class
from HdkApi.Data.Object import TerritoryTypes

class TerritoryTypes_Example():
    def __init__(self):

        territories = [TerritoryTypes.SCEA, TerritoryTypes.SCEE, TerritoryTypes.SCEJ, TerritoryTypes.SCEAsia]
        for territory in territories:
            print (territory)

tt_example = TerritoryTypes_Example()
```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.Uuid

Uuid

Uuid object used to identify an object. To create a new Uuid use `Uuid.Create()`

Namespace: [HdkApi.Data.Object](#)

Assembly: *Home.Data.Scripting.dll*

Methods

`HdkApi.Data.Object.Uuid Uuid.Create( uuid )`

`System.Boolean Uuid.IsValid( uuid )`

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Uuid class
from HdkApi.Data.Object import Uuid

class UuidExample():
    def __init__(self):

        strHelicopterUuid = "45864836-89475616-84123186-13213216"
        isValid = Uuid.IsValid(strHelicopterUuid)
        if isValid:
            HelicopterUuid = Uuid.Create(strHelicopterUuid)
            print "Get to the chopper! %s" % strHelicopterUuid

        strDinosaurUuid = "DINOSAUR-SARENOTA-UUIDVELO-CIRAPTOR"
        isValid = Uuid.IsValid(strDinosaurUuid)
        if isValid == False:
            print "Dinosaurs are extinct!"

ux = UuidExample()

```

Associated Types

- [HdkApi.Data.Object.ObjectManager](#)
- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.Components

Summary

Type
Description
Assembly
ClothingComponent
Provides information about a clothing component
Home.Data.Scripting
HeaderComponent
Header information for an object, used for indexing and general information.
Home.Data.Scripting

FurnitureComponent
Provides information about a furniture component
Home.Data.Scripting
GameLaunchComponent
Provides information about a game_launch component
Home.Data.Scripting

HdkApi.Data.Object.Components.ClothingComponent

ClothingComponent

Provides information about a clothing component

Namespace: [HdkApi.Data.Object.Components](#)

Assembly: *Home.Data.Scripting.dll*

Properties

Name

ClothingType

Clothing

Rigs

Type

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# Import the Object classes
from HdkApi.Data.Object import ObjectManager, Uuid, ComponentTypes

# Import the ClothingComponent class
from HdkApi.Data.Object.Components import ClothingComponent

class ClothingExample():
    def __init__(self):

        # Load an object by its uuid
        uuid = Uuid.Create("00000000-00000000-00000000-00000018")

        try:
            loadedObject = ObjectManager.LoadByUuid(uuid)

            if loadedObject.Components.ContainsKey(ComponentTypes.Clothing):
                # Change the category of the clothing item
                loadedObject.Components[ComponentTypes.Clothing].Clothing = "Hat"

                # Change the rig which the clothing item is designed for
                loadedObject.Components[ComponentTypes.Clothing].Rigs =
                "00000000-00000000-00000010-00000000"

        except Exception, e:
            print "Failed to load object {0}: {1}".format(uuid, e)

```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.Components.FurnitureComponent

FurnitureComponent

Provides information about a furniture component

Namespace: [HdkApi.Data.Object.Components](#)

Assembly: *Home.Data.Scripting.dll*

Properties

Furniture

Type

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# Import the Object classes
from HdkApi.Data.Object import ObjectManager, Uuid, ComponentTypes

# Import the FurnitureComponent class
from HdkApi.Data.Object.Components import FurnitureComponent

class FurnitureExample():
    def __init__(self):

        # Load an object by its uuid
        uuid = Uuid.Create("00000000-00000000-00000000-00000018")

        try:
            loadedObject = ObjectManager.LoadByUuid(uuid)

            if loadedObject.Components.ContainsKey(ComponentTypes.Furniture):
                # Change the category of the furniture item
                loadedObject.Components[ComponentTypes.Furniture].Furniture = "Desk"

        except Exception, e:
            print "Failed to load object {0}: {1}".format(uuid, e)

```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.Components.GameLaunchComponent

GameLaunchComponent

Provides information about a game_launch component

Namespace: [HdkApi.Data.Object.Components](#)

Assembly: *Home.Data.Scripting.dll*

Properties

CommunicationId

TitleId

Type

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# Import the Object classes
from HdkApi.Data.Object import ObjectManager, Uuid, ComponentTypes

# Import the GameLaunchComponent class
from HdkApi.Data.Object.Components import GameLaunchComponent

class GameLaunchExample():
    def __init__(self):

        # Load an object by its uuid
        uuid = Uuid.Create("00000000-00000000-00000000-00000018")

        try:
            loadedObject = ObjectManager.LoadByUuid(uuid)

            if loadedObject.Components.ContainsKey(ComponentTypes.GameLaunch):
                # Change the communication id
                loadedObject.Components[ComponentTypes.GameLaunch].CommunicationId = "COMM_ID"

                # Change the title id
                loadedObject.Components[ComponentTypes.GameLaunch].TitleId = "GAME_TITLE"

        except Exception, e:
            print "Failed to load object {0}: {1}".format(uuid, e)

```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Object.Components.HeaderComponent

HeaderComponent

Header information for an object, used for indexing and general information.

Namespace: [HdkApi.Data.Object.Components](#)

Assembly: *Home.Data.Scripting.dll*

Properties

Name

Description

Author

ObjectVersion

Publisher

Template

TemplateVersion

AgeRatings

HelpText

Lpid

Type

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Object classes
from HdkApi.Data.Object import ObjectManager, HomeObject, ObjectComponent, Uuid, ComponentTypes,
TerritoryTypes

# Import the MessageType class as we need this for the object validation and packaging output
from HdkApi.Data.Reporting import MessageType

# Also import PathUtilities as we use it to get the project path
from HdkApi.Core.Utilities import PathUtilities

class ObjectManagerExample():
    def __init__(self):
        try:
            # Load an object by its uuid
            uuid = Uuid.Create("00000000-00000000-00000000-00000018")
            loadedObject = ObjectManager.LoadByUuid(uuid)

            # Load an object by its object.xml path
            objectPath = PathUtilities.ProjectPath +
"/build/objects/00000000-00000000-00000000-00000018/object.xml"
            loadedObject = ObjectManager.LoadByFilename(objectPath)

        except Exception, e:
            print "Failed to load object {0}: {1}".format(uuid, e)

        # Get the Uuids of all objects which contain either a 'furniture' or an 'active' component
        components = ["furniture", "active"]
        foundObjects = ObjectManager.SearchByComponents(components)

        # Get the Uuids of all objects which contain a resource with the path 'resourcePath'
        resourcePath = "objects/00000000-00000000-00000000-00000018/main.lua"
        foundObjects = ObjectManager.SearchByResource(resourcePath)

        # Print the entitlement_id for the territory 'SCEA'
        print loadedObject.EntitlementId[TerritoryTypes.SCEA]

        # Set the entitlement_id for the territory 'SCEE'
        loadedObject.EntitlementId[TerritoryTypes.SCEE] = "LUA_REWARD"

        # Print the name of each component which loadedObject contains
        for component in loadedObject.Components.Keys:
            print component

        # Try to access the furniture component and set the furniture metadata item
        if loadedObject.Components.ContainsKey(ComponentTypes.Furniture):
            loadedObject.Components[ComponentTypes.Furniture].Furniture = "Chair"

        # Validate an object
        if loadedObject.Validate(self.Output):
            print "{0} - {1} passed validations\n".format(loadedObject.Name, loadedObject.Uuid)
```

```
# Validate and package an object
# The first passed in callback is triggered for validation output
# The second callback is triggered for packager output
if loadedObject.Package(self.Output, self.Output):
    print "{0} - {1} succesfully packaged\n".format(loadedObject.Name, loadedObject.Uuid)

# Example function for handling object validation and packager output
def Output(self, msg, msgType):
    if (msgType == MessageType.INFO):
        print "INFO: {0}\n".format(msg)

    elif (msgType == MessageType.WARNING):
        print "WARNING: {0}\n".format(msg)

    elif (msgType == MessageType.ERROR):
        print "ERROR: {0}\n".format(msg)

    elif (msgType == MessageType.FATAL):
```

```
print "FATAL: {0}\n".format(msg)

objectManagerExample = ObjectManagerExample()
```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.Data.Packager

Summary

Type
Description
Assembly
Packager
The Packager class is used to package objects and scenes
Home.Data.Scripting

HdkApi.Data.Packager.Packager

Packager

The Packager class is used to package objects and scenes

Namespace: [HdkApi.Data.Packager](#)

Assembly: *Home.Data.Scripting.dll*

Methods

System.Boolean Packager.PackageObject(uuid, validatorOutput, packagerOutput)

System.Boolean Packager.PackageScene(scenePath, sceneType, validatorOutput, packagerOutput)

HdkApi.Data.Reporting

Summary

Type
Description
Assembly

MessageType
The types of messages which can be returned through the MessageOutput delegate
Home.Data.Scripting

HdkApi.Data.Reporting.MessageType

MessageType

The types of messages which can be returned through the MessageOutput delegate

Namespace: [HdkApi.Data.Reporting](#)

Assembly: *Home.Data.Scripting.dll*

Fields

INFO

WARNING

ERROR

FATAL

HdkApi.Data.Scene

Summary

Type
Description
Assembly
GameObjectFolder
Provides information about a GameObjectFolder within a Scene
Home.Data.Scripting
GameObject
Provides information about a Game Object within a Scene
Home.Data.Scripting
GameObjectTypes
Lists all of the game object types
Home.Data.Scripting

HomeScene
Provides information about a scene
Home.Data.Scripting
SceneConverter
Provides functions for converting between scene (.scene) and scene project (.scproj) files.
Home.Data.Scripting
SceneTypes
Lists all of the scene types
Home.Data.Scripting

HdkApi.Data.Scene.GameObject

GameObject

Provides information about a Game Object within a Scene

Namespace: [HdkApi.Data.Scene](#)
Assembly: *Home.Data.Scripting.dll*

Properties

Name

Type

Translation

Rotation

Scale

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the SceneProject classes
from HdkApi.Data.Scene import HomeScene, GameObject

# Import the SceneEditor class so we can get the current scene project open in the Scene Editor
from HdkApi.SceneEditor import SceneEditor

# Import the Maths Converters class so we can use the degrees to radians function
from HdkApi.Core.Maths import Converters

class SceneProjectExample():
    def __init__(self):
        # Get the scene project which is currently open in the Scene Editor
        openSceneProj = SceneEditor.CurrentScene

        # Print the scene project's full file path
        print openSceneProj.FilePath

        # Print the name and type of each game object within the scene project
        for gameObject in openSceneProj.AllGameObjects:
            print "{0} : {1}".format(gameObject.Name, gameObject.Type)

        # Find a game object by its name
        spawnPoint = openSceneProj.FindGameObjectByName("Spawn Point")

        if spawnPoint != None:
            # Change the translation, rotation and scale of the spawn point
            spawnPoint.Translation = Vector3(20, 0, 0)
            spawnPoint.Rotation = Vector3(Converters.DegToRad(90), 0, 0)
            spawnPoint.Scale = Vector3(2, 1, 1)

        # Find all game objects of the type 'entityType'
        entities = openSceneProj.FindGameObjectsByType(GameObjectTypes.Entity)

        # Print the name of all entities in the scene project
        for entity in entities:
            print entity.Name

```

Associated Types

- [HdkApi.Data.Scene.HomeScene](#)
- [HdkApi.Data.Scene.GameObjectTypes](#)

HdkApi.Data.Scene.GameObjectFolder

GameObjectFolder

Provides information about a GameObjectFolder within a Scene

Namespace: [HdkApi.Data.Scene](#)

Assembly: *Home.Data.Scripting.dll*

Properties

Name

GameObjects

Folders

Associated Types

- [HdkApi.Data.Scene.HomeScene](#)

HdkApi.Data.Scene.GameObjectTypes

GameObjectTypes

Lists all of the game object types

Namespace: [HdkApi.Data.Scene](#)

Assembly: *Home.Data.Scripting.dll*

Fields

CollisionFile

Entity

EnvironmentMap

EyeEnvironmentMap

ProbeFile

AtmosphereFile

Model

SkinEnvironmentMap

DefaultFurniturePoint

Seat

SeatArea

PictureHook

SpawnPoint

Script

EmbeddedObject

Screen

ClubBulletinScreen

ScreenLinksFile

ArcadeGame

MiniGame

AmbientSound

SphereEmitter

CubeEmitter

ConeEmitter

CylinderEmitter

ConePointSound

CubePointSound

CylinderPointSound

SpherePointSound

SoundBank

SoundPortal

SoundZone

ChangeEventBoxTrigger

ChangeEventTrigger

ParticleEffect

GameObjectGroup

Associated Types

- [HdkApi.Data.Scene.HomeScene](#)
- [HdkApi.Data.Scene.GameObject](#)

HdkApi.Data.Scene.HomeScene

HomeScene

Provides information about a scene

Namespace: [HdkApi.Data.Scene](#)

Assembly: *Home.Data.Scripting.dll*

Methods

HdkApi.Data.Scene.GameObjectFolder FindGameObjectFolderByName(name)

HdkApi.Data.Scene.GameObject FindGameObjectByName(name)

System.Collections.Generic.IEnumerable[HdkApi.Data.Scene.GameObject] FindGameObjectsByType(type)

System.Boolean Validate(validatorOutput, sceneType)

System.Boolean Package(validatorOutput, packagerOutput, sceneType)

Save()

Properties

FilePath

AllGameObjects

GameObjectFolder

Dependencies

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the SceneProject classes
from HdkApi.Data.Scene import HomeScene, GameObject

# Import the SceneEditor class so we can get the current scene project open in the Scene Editor
from HdkApi.SceneEditor import SceneEditor

# Import the Maths Converters class so we can use the degrees to radians function
from HdkApi.Core.Maths import Converters

class SceneProjectExample():
    def __init__(self):
        # Get the scene project which is currently open in the Scene Editor
        openSceneProj = SceneEditor.CurrentScene

        # Print the scene project's full file path
        print openSceneProj.FilePath

        # Print the name and type of each game object within the scene project
        for gameObject in openSceneProj.AllGameObjects:
            print "{0} : {1}".format(gameObject.Name, gameObject.Type)

        # Find a game object by its name
        spawnPoint = openSceneProj.FindGameObjectByName("Spawn Point")

        if spawnPoint != None:
            # Change the translation, rotation and scale of the spawn point
            spawnPoint.Translation = Vector3(20, 0, 0)
            spawnPoint.Rotation = Vector3(Converters.DegToRad(90), 0, 0)
            spawnPoint.Scale = Vector3(2, 1, 1)

        # Find all game objects of the type 'entityType'
        entities = openSceneProj.FindGameObjectsByType(GameObjectTypes.Entity)

        # Print the name of all entities in the scene project
        for entity in entities:
            print entity.Name

```

Associated Types

- [HdkApi.Data.Scene.GameObject](#)
- [HdkApi.Data.Scene.GameObjectTypes](#)

HdkApi.Data.Scene.SceneConverter

SceneConverter

Provides functions for converting between scene (.scene) and scene project (.scproj) files.

Namespace: [HdkApi.Data.Scene](#)

Assembly: *Home.Data.Scripting.dll*

Methods

System.String SceneConverter.SceneToProject(sceneToConvert, forceOverwrite)

System.String SceneConverter.ProjectToScene(projectToConvert, forceOverwrite)

System.String SceneConverter.GetSceneVersion(sceneToCheck)

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the SceneConverter class
from HdkApi.Data.Scene import SceneConverter

class SceneConverterExample():
    def __init__(self):

        # Convert a scene to a project
        scprojFile = SceneConverter.SceneToProject("D:\foo\bar\demo.scene", True)

        # Convert a project to a scene
        sceneFile = SceneConverter.ProjectToScene("D:\foo\bar\demo.scproj", True)

        # Get a scenes version number
        version = SceneConverter.GetSceneVersion("D:\foo\bar\demo.scene")
```

HdkApi.Data.Scene.SceneTypes

SceneTypes

Lists all of the scene types

Namespace: [HdkApi.Data.Scene](#)

Assembly: *Home.Data.Scripting.dll*

Fields

Public

Personal

ClubHouse

Game16

Game32

Game60

PublicNoVideo

Video

PersonalVideo

PersonalNoVideo

Associated Types

- [HdkApi.Data.Scene.HomeScene](#)

HdkApi.Data.Scene.GameObjects

Summary

Type
Description
Assembly
GameObjectGroup
Provides information about a Game Object Group
Home.Data.Scripting

HdkApi.Data.Scene.GameObjects.GameObjectGroup

GameObjectGroup

Provides information about a Game Object Group

Namespace: [HdkApi.Data.Scene.GameObjects](#)

Assembly: *Home.Data.Scripting.dll*

Properties

GameObjects

Name

Type

Translation

Rotation

Scale

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.60.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the SceneProject classes
from HdkApi.Data.Scene import HomeScene, GameObject

# Import the SceneEditor class so we can get the current scene project open in the Scene Editor
from HdkApi.SceneEditor import SceneEditor

# Import the Maths Converters class so we can use the degrees to radians function
from HdkApi.Core.Maths import Converters

class SceneProjectExample():
    def __init__(self):
        # Get the scene project which is currently open in the Scene Editor
        openSceneProj = SceneEditor.CurrentScene

        # Print the scene project's full file path
        print openSceneProj.FilePath

        # Print the name and type of each game object within the scene project
        for gameObject in openSceneProj.AllGameObjects:
            print "{0} : {1}".format(gameObject.Name, gameObject.Type)

        # Find a game object by its name
        spawnPoint = openSceneProj.FindGameObjectByName("Spawn Point")

        if spawnPoint != None:
            # Change the translation, rotation and scale of the spawn point
            spawnPoint.Translation = Vector3(20, 0, 0)
            spawnPoint.Rotation = Vector3(Converters.DegToRad(90), 0, 0)
            spawnPoint.Scale = Vector3(2, 1, 1)

        # Find all game objects of the type 'entityType'
        entities = openSceneProj.FindGameObjectsByType(GameObjectTypes.Entity)

        # Print the name of all entities in the scene project
        for entity in entities:
            print entity.Name

```

Associated Types

- [HdkApi.Data.Scene.HomeScene](#)

HdkApi.Data.Validator

Summary

Type

Description

Assembly

Validator

The Validator class is used to validate objects and scenes
Home.Data.Scripting

HdkApi.Data.Validator.Validator

Validator

The Validator class is used to validate objects and scenes

Namespace: [HdkApi.Data.Validator](#)

Assembly: *Home.Data.Scripting.dll*

Methods

System.Boolean Validator.ValidateObject(uuid, validatorOutput)

System.Boolean Validator.ValidateScene(scenePath, sceneType, validatorOutput)

HdkApi.ObjectEditor

Summary

Type
Description
Assembly
ObjectEditor
Provides access to core functionality of the Object Editor, such as events for when objects are opened.
Home.Editors.ObjectEditor.Scripting

HdkApi.ObjectEditor.ObjectEditor

ObjectEditor

Provides access to core functionality of the Object Editor, such as events for when objects are opened.

Namespace: [HdkApi.ObjectEditor](#)

Assembly: *Home.Editors.ObjectEditor.Scripting.dll*

Methods

ObjectEditor.OpenObjectByUuid(uuid)

ObjectEditor.CloseEditor()

System.Collections.Generic.IEnumerable[HdkApi.Data.Object.HomeObject] ObjectEditor.GetOpenObjects()

Events

OnSaving

OnSaved

OnOpened

OnPackaging

OnPackaged

Delegates

ObjectAction

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the ObjectEditor class
from HdkApi.ObjectEditor import ObjectEditor

# Import the HomeObject class
from HdkApi.Data.Object import HomeObject

class ObjectEditorExample():
    def __init__(self):
        # Hook up to the OnOpened event called when an object is opened in the Object Editor
        ObjectEditor.OnOpened += self.onOpened

        # Hook up to the OnSaving event called when an object is about to be saved
        ObjectEditor.OnSaving += self.onSaving

        # Hook up to the OnSaved event called when an object has been successfully saved
        ObjectEditor.OnSaved += self.onSaved

        # Hook up to the OnPackaging event called when an object is about to be packaged
        ObjectEditor.OnPackaging += self.onPackaging

        # Hook up to the OnPackaged event called when an object has been successfully packaged
        ObjectEditor.OnPackaged += self.onPackaged

        # Example of a callback function triggered when an object is opened in the Object Editor
        def onOpened(self, object):
            print "Opened: {0}".format(object.FilePath)

        # Example of a callback function triggered when an object is about to be saved in the Object
        Editor
        def onSaving(self, object):
            print "Saving: {0}".format(object.FilePath)

        # Example of a callback function triggered when an object has been successfully saved in the
        Object Editor
        def onSaved(self, object):
            print "Saved: {0}".format(object.FilePath)

        # Example of a callback function triggered when an object is about to be packaged in the Object
        Editor
        def onPackaging(self, object):
            print "Packaging: {0}".format(object.FilePath)

        # Example of a callback function triggered when an object has been successfully packaged in the
        Object Editor
        def onPackaged(self, object):
            print "Packaged: {0}".format(object.FilePath)

```

Associated Types

- [HdkApi.Data.Object.HomeObject](#)

HdkApi.ObjectEditor.UserInterface

Summary

Type
Description
Assembly
Menu
Menu represents a menu in the Object Editor UI, new items can be added to the menu by calling the AddMenuItem function, a new Menu can be created through MenuManager.CreateMenu
Home.Editors.ObjectEditor.Scripting
MenuItem
MenuItem represents an item within a Menu in the Object Editor UI, it provides access to properties about the menu item such as what icon it uses and whether it has a button associated with it, for a menu item to be visible it must be added to a Menu through AddMenuItem, to create a new MenuItem call MenuManager.CreateMenuItem
Home.Editors.ObjectEditor.Scripting
MenuManager
MenuManager is used to add new menus and menu items to the Object Editor UI, to add a new menu call CreateMenu, to add a new menu item call CreateMenuItem and then pass the returned MenuItem to a menu's AddMenuItem function
Home.Editors.ObjectEditor.Scripting

HdkApi.ObjectEditor.UserInterface.Menu

Menu

Menu represents a menu in the Object Editor UI, new items can be added to the menu by calling the AddMenuItem function, a new Menu can be created through MenuManager.CreateMenu

Namespace: [HdkApi.ObjectEditor.UserInterface](#)
Assembly: *Home.Editors.ObjectEditor.Scripting.dll*

Methods

AddMenuItem(menuItem)

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the UserInterface classes
from HdkApi.SceneEditor.UserInterface import MenuManager, Menu, MenuItem

# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("Example Menu")

        # Create a new MenuItem
        itemDisplayText = "Example Item"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Example menu item used in the UI Example"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.M
        menuItem.HasToolBarButton = True

        # Hook up the execute event to set what happens when the menu item is pressed
        menuItem.Execute += self.exampleExecute

        # Hook up the can execute event to set whether the menu item is enabled
        menuItem.CanExecute += self.exampleCanExecute

        # Add the MenuItem to the example menu
        menu.AddMenuItem(menuItem)

    def exampleCanExecute(self):
        return True

    def exampleExecute(self, sender, args):
        print("Example executed")

example = UiExample()

```

Associated Types

- [HdkApi.ObjectEditor.UserInterface.MenuManager](#)
- [HdkApi.ObjectEditor.UserInterface.MenuItem](#)

HdkApi.ObjectEditor.UserInterface.MenuItem

MenuItem

MenuItem represents an item within a Menu in the Object Editor UI, it provides access to properties about the menu item such as what icon it uses and whether it has a button associated with it, for a menu item to be visible it must be added to a Menu through AddMenuItem, to create a new MenuItem call MenuManager.CreateMenuItem

Namespace: [HdkApi.ObjectEditor.UserInterface](#)
Assembly: *Home.Editors.ObjectEditor.Scripting.dll*

Properties

DisplayText

Description

IconPath

Shortcut

Toggleable

HasToolBarButton

Events

Execute

CanExecute

Delegates

CanExecuteHandler

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the UserInterface classes
from HdkApi.SceneEditor.UserInterface import MenuManager, Menu, MenuItem

# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("Example Menu")

        # Create a new MenuItem
        itemDisplayText = "Example Item"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Example menu item used in the UI Example"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.M
        menuItem.HasToolBarButton = True

        # Hook up the execute event to set what happens when the menu item is pressed
        menuItem.Execute += self.exampleExecute

        # Hook up the can execute event to set whether the menu item is enabled
        menuItem.CanExecute += self.exampleCanExecute

        # Add the MenuItem to the example menu
        menu.AddMenuItem(menuItem)

    def exampleCanExecute(self):
        return True

    def exampleExecute(self, sender, args):
        print("Example executed")

example = UiExample()

```

Associated Types

- [HdkApi.ObjectEditor.UserInterface.MenuManager](#)
- [HdkApi.ObjectEditor.UserInterface.Menu](#)

HdkApi.ObjectEditor.UserInterface.MenuManager

MenuManager

MenuManager is used to add new menus and menu items to the Object Editor UI, to add a new menu call CreateMenu, to add a new menu item call CreateMenuItem and then pass the returned MenuItem to a menu's AddMenuItem function

Namespace: [HdkApi.ObjectEditor.UserInterface](#)
Assembly: *Home.Editors.ObjectEditor.Scripting.dll*

Methods

HdkApi.ObjectEditor.UserInterface.Menu MenuManager.CreateMenu(name)

HdkApi.ObjectEditor.UserInterface.Menu MenuManager.GetMenu(name)

HdkApi.ObjectEditor.UserInterface.MenuItem MenuManager.CreateMenuItem(displayText)

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.55.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the UserInterface classes
from HdkApi.SceneEditor.UserInterface import MenuManager, Menu, MenuItem

# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("Example Menu")

        # Create a new MenuItem
        itemDisplayText = "Example Item"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Example menu item used in the UI Example"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.M
        menuItem.HasToolBarButton = True

        # Hook up the execute event to set what happens when the menu item is pressed
        menuItem.Execute += self.exampleExecute

        # Hook up the can execute event to set whether the menu item is enabled
        menuItem.CanExecute += self.exampleCanExecute

        # Add the MenuItem to the example menu
        menu.AddMenuItem(menuItem)

    def exampleCanExecute(self):
        return True

    def exampleExecute(self, sender, args):
        print("Example executed")

example = UiExample()
```

Associated Types

- [HdkApi.ObjectEditor.UserInterface.Menu](#)
- [HdkApi.ObjectEditor.UserInterface.MenuItem](#)

HdkApi.RepertoireEditor

Summary

Type
Description
Assembly
Action
An action defines one off animations within repertoire
Home.Editors.RepertoireEditor.Scripting
Behaviour
A behaviour defines repetitive behaviour in repertoire
Home.Editors.RepertoireEditor.Scripting
EaseCurveType
Determines how a curve blends over time
Home.Editors.RepertoireEditor.Scripting
EaseCurve
Describes how a blend in or out of a state happens over time
Home.Editors.RepertoireEditor.Scripting
Locomotion
Locomotion defines movement within repertoires
Home.Editors.RepertoireEditor.Scripting
LocomotionState
Locomotion state module used to define locomotions in repertoire
Home.Editors.RepertoireEditor.Scripting
RepertoireRegister
Home.Editors.RepertoireEditor.Scripting
BooleanRegister

A boolean value register
Home.Editors.RepertoireEditor.Scripting
IntegerRegister
An integer value register
Home.Editors.RepertoireEditor.Scripting
NumberRegister
A number value register
Home.Editors.RepertoireEditor.Scripting
StringRegister
A string value register
Home.Editors.RepertoireEditor.Scripting
Repertoire
Repertoire class used to create and access repertoire modules
Home.Editors.RepertoireEditor.Scripting
RepertoireEditor
The main class for repertoire editor, use it to access currently open repertoires
Home.Editors.RepertoireEditor.Scripting
RepertoireEvent
Home.Editors.RepertoireEditor.Scripting
RepertoireTargetRig
The rig, emotes and repertoire resources referenced in the repertoire component.
Home.Editors.RepertoireEditor.Scripting
State
A state can be part of a behaviour or an action

Home.Editors.RepertoireEditor.Scripting
Transition
Create a transition between 2 animations by specifying a transition animation
Home.Editors.RepertoireEditor.Scripting

HdkApi.RepertoireEditor.Action

Action

An action defines one off animations within repertoire

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Conditions.Condition GetCondition()

HdkApi.RepertoireEditor.Conditions.Condition CreateCondition(name)

SetDuration(duration)

System.Object GetDuration()

HdkApi.RepertoireEditor.State GetInitialState()

SetInitialState(state)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Outputs.Output] GetOutputs()

HdkApi.RepertoireEditor.Outputs.Output CreateOutput(name)

System.Collections.Generic.List[HdkApi.RepertoireEditor.State] GetStates()

HdkApi.RepertoireEditor.State CreateState(name)

Delete()

Properties

Layer

Blocking

Priority

States

Attributes

Position

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateAction():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        action = rep.CreateAction("Action")
        if(action):
            #set some properties
            action.Name = "Action"
            action.Layer = "my_layer"
            action.Blocking = True
            action.Priority = 1
            action.Position = Point(500,100)

            #set the duration of the action
            action.SetDuration(5)

            #set some attributes
            action.Attributes["something"] = 10

            #create a condition for this action
            condition = action.CreateCondition("condition")
            if(condition):
                print("created condition, you may wish to use this to add some logic")
                condition.Position = Point(350, 100)

            #create a state for this action
            state = action.CreateState("State")
            if(state):
                print("created state, you may wish to add animations to this state")
                state.Position = Point(500, 100)

            #set this state as the initial state for the action
            action.SetInitialState(state)

            #create an output for this action
            output = action.CreateOutput("Output")
            if(output):
                print("created output, you may wish to use this to define behaviour on entry and exit
to this action")
                output.Position = Point(500, 200)

            #get the condition back from the action
            theCondition = action.GetCondition()
            if(theCondition):
                print("got the condition of this action")

            #get the list of states from this action
            theStates = action.GetStates()
            if(theStates):
                print("got the states of this action")

            #get the list of outputs from this action
            theOutputs = action.GetOutputs()
            if(theOutputs):
                print("got the list of outputs of this action")

    CreateAction()

```

HdkApi.RepertoireEditor.Behaviour

Behaviour

A behaviour defines repetitive behaviour in repertoire

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Conditions.Condition GetCondition()

HdkApi.RepertoireEditor.Conditions.Condition CreateCondition(name)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Outputs.Output] GetOutputs()

HdkApi.RepertoireEditor.Outputs.Output CreateOutput(name)

HdkApi.RepertoireEditor.State GetInitialState()

SetInitialState(state)

System.Collections.Generic.List[HdkApi.RepertoireEditor.State] GetStates()

HdkApi.RepertoireEditor.State CreateState(name)

System.Collections.Generic.List[HdkApi.RepertoireEditor.LocomotionState] GetLocomotionStates()

HdkApi.RepertoireEditor.LocomotionState CreateLocomotionState(name)

Delete()

Properties

States

Attributes

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateBehaviour():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        behaviour = rep.CreateBehaviour("Behaviour")
        if(behaviour):
            #set some properties
            behaviour.Name = "Behaviour"
            behaviour.Position = Point(500,100)

            #set some attributes
            behaviour.Attributes["something"] = 10
```

```

#create a condition for this behaviour
condition = behaviour.CreateCondition("condition")
if(condition):
    print("created condition, you may wish to use this to add some logic")
    condition.Position = Point(350,100)

#create a state for this behaviour
state = behaviour.CreateState("State")
if(state):
    print("created state, you may wish to add animations to this state")
    state.Position = Point(500,100)

#set the initial state
behaviour.SetInitialState(state)

#create a locomotion state for this behaviour
locomotionState = behaviour.CreateLocomotionState("Locomotion_State")
if(locomotionState):
    print("created locomotion state, you may wish to add animations to this state")
    locomotionState.Position = Point(500,200)

#create an output for this behaviour
output = behaviour.CreateOutput("Output")
if(output):
    print("created output, you may wish to use this to define behaviour on entry and exit
to this behaviour")
    output.Position = Point(500,300)

#get the conditon back from the behaviour
theCondition = behaviour.GetCondition()
if(theCondition):
    print("got the condition of this behaviour")

#get the list of states from this behaviour
theStates = behaviour.GetStates()
if(theStates):
    print("got the states of this behaviour")

#get the list of states from this behaviour
theLocomotionStates = behaviour.GetLocomotionStates()
if(theLocomotionStates):
    print("got the locomotion states of this behaviour")

#get the list of outputs from this behaviour
theOutputs = behaviour.GetOutputs()
if(theOutputs):

```

```
print("got the list of outputs of this behaviour")  
CreateBehaviour()
```

HdkApi.RepertoireEditor.BooleanRegister

BooleanRegister

A boolean value register

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

Properties

Name

RepertoireWrite

RepertoireRead

LuaRead

LuaWrite

IsCore

HdkApi.RepertoireEditor.EaseCurve

EaseCurve

Describes how a blend in or out of a state happens over time

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Constructors

EaseCurve(curveType, duration)

Methods

SetDuration(duration)

System.Object GetDuration()

HdkApi.RepertoireEditor.EaseCurveType GetCurveType()

SetCurveType(curveType)

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateEaseCurve():
    #create an ease curve
    easeCurve = EaseCurve(EaseCurveType.linear, 5.0)

    if(easeCurve):
        print("created ease curve")

        #get the type
        type = easeCurve.GetCurveType()
        print("type = " + str(type))
        #get the duration
        duration = easeCurve.GetDuration()
        print("duration = " + str(duration))

        #set the duration
        easeCurve.SetDuration(2.5)

        #set the type
        easeCurve.SetCurveType(EaseCurveType.s_curve)

CreateEaseCurve()

```

HdkApi.RepertoireEditor.EaseCurveType

EaseCurveType

Determines how a curve blends over time

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Fields

linear

s_curve

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateEaseCurve():
    #create an ease curve
    easeCurve = EaseCurve(EaseCurveType.linear, 5.0)

    if(easeCurve):
        print("created ease curve")

        #get the type
        type = easeCurve.GetCurveType()
        print("type = " + str(type))
        #get the duration
        duration = easeCurve.GetDuration()
        print("duration = " + str(duration))

        #set the duration
        easeCurve.SetDuration(2.5)

        #set the type
        easeCurve.SetCurveType(EaseCurveType.s_curve)

CreateEaseCurve()

```

HdkApi.RepertoireEditor.IntegerRegister

IntegerRegister

An integer value register

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

Properties

Min

Max

Name

RepertoireWrite

RepertoireRead

LuaRead

LuaWrite

IsCore

HdkApi.RepertoireEditor.Locomotion

Locomotion

Locomotion defines movement within repertoires

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Animation.SkeletalAnimation GetSkeletalAnimation()

HdkApi.RepertoireEditor.Animation.SkeletalAnimation CreateSkeletalAnimation(name)

HdkApi.RepertoireEditor.Animation.BlendTreeAnimation GetBlendTreeAnimation()

HdkApi.RepertoireEditor.Animation.BlendTreeAnimation CreateBlendTreeAnimation(name)

Delete()

Properties

Speed

Threshold

Position

Name

HdkApi.RepertoireEditor.LocomotionState

LocomotionState

Locomotion state module used to define locomotions in repertoire

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Conditions.Condition GetCondition()

HdkApi.RepertoireEditor.Conditions.Condition CreateCondition(name)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Locomotion] GetLocomotions()

HdkApi.RepertoireEditor.Locomotion CreateLocomotion(name)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Animation.SkeletalAnimation] GetSkeletalAnimations()

System.Collections.Generic.List[HdkApi.RepertoireEditor.Animation.BlendTreeAnimation] GetBlendTreeAnimations()

SetDuration(duration)

System.Object GetDuration()

HdkApi.RepertoireEditor.EaseCurve GetEaseIn()

SetEaseIn(easeCurve)

HdkApi.RepertoireEditor.EaseCurve GetEaseOut()

SetEaseOut(easeCurve)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Outputs.Output] GetOutputs()

HdkApi.RepertoireEditor.Outputs.Output CreateOutput(name)

Delete()

Properties

SkeletalAnimations

BlendTreeAnimations

Position

Name

HdkApi.RepertoireEditor.NumberRegister

NumberRegister

A number value register

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

Properties

Min

Max

Name

RepertoireWrite

RepertoireRead

LuaRead

LuaWrite

IsCore

HdkApi.RepertoireEditor.Repertoire

Repertoire

Repertoire class used to create and access repertoire modules

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

System.Boolean Save()

HdkApi.Core.Collections.HdkDictionary[System.String,System.Collections.Generic.List[System.String]] Validate()

System.Boolean Export()

System.Collections.Generic.List[HdkApi.RepertoireEditor.Action] GetActions()

HdkApi.RepertoireEditor.Action CreateAction(name)

HdkApi.RepertoireEditor.Behaviour CreateBehaviour(name)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Behaviour] GetBehaviours()
System.Collections.Generic.List[HdkApi.RepertoireEditor.Transition] GetTransitions()
HdkApi.RepertoireEditor.Emotes.RegionalEmotes CreateRegionalEmotes(name)
System.Collections.Generic.List[HdkApi.RepertoireEditor.Emotes.RegionalEmotes] GetRegionalEmotes()
HdkApi.RepertoireEditor.Emotes.GlobalEmotes CreateGlobalEmotes(name)
HdkApi.RepertoireEditor.Emotes.GlobalEmotes GetGlobalEmotes()
HdkApi.RepertoireEditor.Animation.BlendTree CreateBlendTree(name)
System.Collections.Generic.List[HdkApi.RepertoireEditor.Animation.BlendTree] GetBlendTrees()
HdkApi.Data.Object.HomeObject GetObject()
System.Collections.Generic.List[HdkApi.Data.Object.ObjectResource] GetAnimationResources()
HdkApi.RepertoireEditor.RepertoireRegister CreateStringRegister(name)
HdkApi.RepertoireEditor.RepertoireRegister CreateBooleanRegister(name)
HdkApi.RepertoireEditor.RepertoireRegister CreateNumberRegister(name)
HdkApi.RepertoireEditor.RepertoireRegister CreateIntegerRegister(name)

Properties

RepertoireCircuitFilepath

RepertoireFilepath

EmotesFilepath

Actions

Behaviours

Transitions

BlendTrees

Events

Registers

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
import System

def UseRepertoire():
    repertoire = RepertoireEditor.CurrentRepertoire
    if(repertoire):
        action = repertoire.CreateAction("action")
        if(action):
            action.Position = Point(500,100)
            print("created action")

        behaviour = repertoire.CreateBehaviour("behaviour")
        if(behaviour):
            behaviour.Position = Point(500,100)
            print("created behaviour")

        emotes = repertoire.CreateGlobalEmotes("global_emotes")
        if(emotes):
            emotes.Position = Point(500,100)
            print("created emotes")

        object = repertoire.GetObject()
        if(object):
            print "Object Name : " + object.Name

        animationResources = repertoire.GetAnimationResources()
        for resource in animationResources:
            print "Object Resource : " + resource.Name

        print("Save")
        repertoire.Save()

    #you could validate your repertoire like this:
    print("Validate")
    results = repertoire.Validate()

    for resultLevel in results.Keys:
        for resultValue in results[resultLevel]:
            print(resultLevel + " : " + resultValue)

    #or handle the exceptions thrown from the validation that happens when you export.
    try:
        repertoire.Export()
    except System.Exception, e:
        print(e.Message)
        print(e.InnerException)

UseRepertoire()

```

HdkApi.RepertoireEditor.RepertoireEditor

RepertoireEditor

The main class for repertoire editor, use it to access currently open repertoires

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Repertoire RepertoireEditor.Open(homeObject, targetRig)

RepertoireEditor.Close(repertoire)

RepertoireEditor.CloseAll()

System.Collections.Generic.List[HdkApi.RepertoireEditor.Repertoire] RepertoireEditor.GetOpenRepertoires()

System.Collections.Generic.List[HdkApi.RepertoireEditor.RepertoireTargetRig] RepertoireEditor.GetRepertoireTargetRigs(homeObject)

Properties

SynchronizationObject

CurrentRepertoire

Events

OnSaving

OnSaved

OnOpened

OnExporting

OnExported

Delegates

RepertoireAction

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
class TestRepertoireEditor():
    def __init__(self):
        RepertoireEditor.OnExporting += self.exporting
        RepertoireEditor.OnExported += self.exported
        RepertoireEditor.OnSaving += self.saving
        RepertoireEditor.OnSaved += self.saved
        RepertoireEditor.OnOpened += self.opened

    def exporting(self, args):
        print("exporting")

    def exported(self, args):
        print("exported")

    def saving(self, args):
        print("saving")

    def saved(self, args):
        print("saved")

    def opened(self, args):
        print("opened")

    def openARepertoireFromAnObject(self):
        components = ["repertoire"]
        foundObjects = ObjectManager.SearchByComponents(components)
        if(len(foundObjects) > 0):
            #we have a loaded object that contains a repertoire component
            obj = ObjectManager.LoadByUuid(foundObjects[0])

            #we need to get the supported target rigs of the object, a mini game will have male and
            female, a full body suit may only have one.
            supportedTargetRigs = RepertoireEditor.GetRepertoireTargetRigs(obj)
            if(len(supportedTargetRigs) > 0):
                RepertoireEditor.Open(obj, supportedTargetRigs[0])

example = TestRepertoireEditor()
example.openARepertoireFromAnObject()

```

HdkApi.RepertoireEditor.RepertoireEvent

RepertoireEvent

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Properties

Name

Values

HdkApi.RepertoireEditor.RepertoireRegister

RepertoireRegister

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

Properties

Name

RepertoireWrite

RepertoireRead

LuaRead

LuaWrite

IsCore

HdkApi.RepertoireEditor.RepertoireTargetRig

RepertoireTargetRig

The rig, emotes and repertoire resources referenced in the repertoire component.

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Properties

Name

RepertoireResource

EmotesResource

HdkApi.RepertoireEditor.State

State

A state can be part of a behaviour or an action

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Conditions.Condition GetCondition()

HdkApi.RepertoireEditor.Conditions.Condition CreateCondition(name)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Animation.SkeletalAnimation] GetSkeletalAnimations()

```

HdkApi.RepertoireEditor.Animation.SkeletalAnimation CreateSkeletalAnimation( name )

System.Collections.Generic.List[HdkApi.RepertoireEditor.Animation.BlendTreeAnimation] GetBlendTreeAnimations( )

HdkApi.RepertoireEditor.Animation.BlendTreeAnimation CreateBlendTreeAnimation( name )

System.Collections.Generic.List[HdkApi.RepertoireEditor.Outputs.Output] GetOutputs( )

HdkApi.RepertoireEditor.Outputs.Output CreateOutput( name )

SetDuration( duration )

System.Object GetDuration( )

HdkApi.RepertoireEditor.EaseCurve GetEaseIn( )

SetEaseIn( easeCurve )

HdkApi.RepertoireEditor.EaseCurve GetEaseOut( )

SetEaseOut( easeCurve )

Delete( )

```

Properties

```

SkeletalAnimations

BlendTreeAnimations

Position

Name

```

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateState():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        action = rep.CreateAction("Action")
        if(action):
            #set some properties
            action.Name = "Action"
            action.Position = Point(500,100)

            #create a state for this action
            state = action.CreateState("State")
            if(state):
                state.Position = Point(500,100)

                condition = state.CreateCondition("Condition")
                if(condition):
                    condition.Position = Point(350,100)

                animation = state.CreateSkeletalAnimation("Animation")
                if(animation):
                    animation.Position = Point(500,100)

    CreateState()

```

HdkApi.RepertoireEditor.StringRegister

StringRegister

A string value register

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

Properties

Values

Name

RepertoireWrite

RepertoireRead

LuaRead

LuaWrite

IsCore

HdkApi.RepertoireEditor.Transition

Transition

Create a transition between 2 animations by specifying a transition animation

Namespace: [HdkApi.RepertoireEditor](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Remarks



Transitions can be achieved by other means

Methods

HdkApi.Data.Object.ObjectResource GetToAnimation()

HdkApi.Data.Object.ObjectResource GetFromAnimation()

HdkApi.RepertoireEditor.Animation.SkeletalAnimation GetSkeletalAnimation()

Delete()

Properties

Position

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def GetTransitions():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        transitions = rep.GetTransitions()
        for transition in transitions:
            print(transition.Name)
            fromAnimationResource = transition.GetFromAnimation()
            if(fromAnimationResource):
                print("From: " + fromAnimationResource.Name)

            transitionAnimationModule = transition.GetSkeletalAnimation()

            if(transitionAnimationModule):
                transitionAnimationResource = transitionAnimationModule.GetResource()
                if(transitionAnimationResource):
                    print("Via: " + transitionAnimationResource.Name)

            toAnimationResource = transition.GetToAnimation()
            if(toAnimationResource):
                print("To: " + toAnimationResource.Name)

    GetTransitions()

```

HdkApi.RepertoireEditor.Animation

Summary

Type
Description
Assembly
BlendTreeAnimation
Blend tree animation used for referencing blend trees
Home.Editors.RepertoireEditor.Scripting
BlendNode
Blends up to 8 animations or blends together to produce one output
Home.Editors.RepertoireEditor.Scripting
BlendTree

A circuit of blend node animations
Home.Editors.RepertoireEditor.Scripting
SkeletalAnimation
Skeletal animation used for referencing animation object resources
Home.Editors.RepertoireEditor.Scripting

HdkApi.RepertoireEditor.Animation.BlendNode

BlendNode

Blends up to 8 animations or blends together to produce one output

Namespace: [HdkApi.RepertoireEditor.Animation](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.Core.Collections.HdkDictionary[System.Int32,HdkApi.RepertoireEditor.Animation.BlendNode] GetBlendNodes()

HdkApi.RepertoireEditor.Animation.BlendNode GetBlendNode(index)

HdkApi.RepertoireEditor.Animation.BlendNode CreateBlendNode(name, index)

HdkApi.Core.Collections.HdkDictionary[System.Int32,HdkApi.RepertoireEditor.Animation.SkeletalAnimation] GetSkeletalAnimations()

HdkApi.RepertoireEditor.Animation.SkeletalAnimation GetSkeletalAnimation(index)

HdkApi.RepertoireEditor.Animation.SkeletalAnimation CreateSkeletalAnimation(name, index)

SetRegister(register)

HdkApi.RepertoireEditor.RepertoireRegister GetRegister()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateBlendTree():
    blendTree = repertoire.CreateBlendTree(name)
    if(blendTree):
        root = blendTree.GetRootBlendNode()
        if(root):
            child1 = root.CreateBlendNode(0)
            anim0 = child1.CreateSkeletalAnimation("0", 0)
            anim1 = child1.CreateSkeletalAnimation("1", 1)

            child2 = root.CreateBlendNode(1)
            anim2 = child2.CreateSkeletalAnimation("0", 0)
            anim3 = child2.CreateSkeletalAnimation("1", 1)

CreateBlendTree()
```

HdkApi.RepertoireEditor.Animation.BlendTree

BlendTree

A circuit of blend node animations

Namespace: [HdkApi.RepertoireEditor.Animation](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Animation.BlendNode GetBlendNode()

HdkApi.RepertoireEditor.Animation.BlendNode CreateBlendNode(name)

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateBlendTree():
    blendTree = repertoire.CreateBlendTree(name)
    if(blendTree):
        root = blendTree.GetRootBlendNode()
        if(root):
            child1 = root.CreateBlendNode(0)
            anim0 = child1.CreateSkeletalAnimation("0", 0)
            anim1 = child1.CreateSkeletalAnimation("1", 1)

            child2 = root.CreateBlendNode(1)
            anim2 = child2.CreateSkeletalAnimation("0", 0)
            anim3 = child2.CreateSkeletalAnimation("1", 1)

CreateBlendTree()
```

HdkApi.RepertoireEditor.Animation.BlendTreeAnimation

BlendTreeAnimation

Blend tree animation used for referencing blend trees

Namespace: [HdkApi.RepertoireEditor.Animation](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Animation.BlendTree GetBlendTree()

SetBlendTree(blendTree)

Delete()

Properties

Position

Name

HdkApi.RepertoireEditor.Animation.SkeletalAnimation

SkeletalAnimation

Skeletal animation used for referencing animation object resources

Namespace: [HdkApi.RepertoireEditor.Animation](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.Data.Object.ObjectResource GetResource()

SetResource(resource)

Delete()

Properties

OverrideDuration

Position

Name

HdkApi.RepertoireEditor.Conditions

Summary

Type
Description
Assembly
And
Performs an and operation on inputs
Home.Editors.RepertoireEditor.Scripting
RandomValue
A random value from a selection or range
Home.Editors.RepertoireEditor.Scripting
CompareCurrentBehaviour
Compare a behaviour to the behaviour which is currently playing
Home.Editors.RepertoireEditor.Scripting
CompareCurrentNetworkAge
Compare the current network age to a number
Home.Editors.RepertoireEditor.Scripting
CompareNetworkAge
Compare the network age to a number
Home.Editors.RepertoireEditor.Scripting
ComparePreviousBehaviour

Compare the previous behaviour to behaviour
Home.Editors.RepertoireEditor.Scripting
ComparePreviousState
Compare the previous state to a state
Home.Editors.RepertoireEditor.Scripting
CompareRegister
Compare a register to a value
Home.Editors.RepertoireEditor.Scripting
CompareStateAge
Compare the current state age to a value
Home.Editors.RepertoireEditor.Scripting
ComparisonType
An enumeration of comparison types
Home.Editors.RepertoireEditor.Scripting
Condition
Define condition logic for use with behaviours, actions and states
Home.Editors.RepertoireEditor.Scripting
Or
Performs an or operation on inputs
Home.Editors.RepertoireEditor.Scripting

HdkApi.RepertoireEditor.Conditions.And

And

Performs an and operation on inputs

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

System.Collections.Generic.List[System.Object] GetInputs()

HdkApi.RepertoireEditor.Conditions.CompareCurrentBehaviour CreateCompareCurrentBehaviour()

HdkApi.RepertoireEditor.Conditions.CompareCurrentNetworkAge CreateCompareCurrentNetworkAge()

HdkApi.RepertoireEditor.Conditions.CompareNetworkAge CreateCompareNetworkAge()

HdkApi.RepertoireEditor.Conditions.ComparePreviousBehaviour CreateComparePreviousBehaviour()

HdkApi.RepertoireEditor.Conditions.ComparePreviousState CreateComparePreviousState()

HdkApi.RepertoireEditor.Conditions.CompareRegister CreateCompareRegister()

HdkApi.RepertoireEditor.Conditions.CompareStateAge CreateCompareStateAge()

HdkApi.RepertoireEditor.Conditions.And CreateAnd()

HdkApi.RepertoireEditor.Conditions.Or CreateOr()

HdkApi.RepertoireEditor.Conditions.RandomValue CreateRandomValue()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.CompareCurrentBehaviour

CompareCurrentBehaviour

Compare a behaviour to the behaviour which is currently playing

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetComparison(comparisonType, behaviour)

IronPython.Runtime.PythonTuple GetComparison()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.CompareCurrentNetworkAge

CompareCurrentNetworkAge

Compare the current network age to a number

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetComparison(comparisonType, value)

IronPython.Runtime.PythonTuple GetComparison()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.CompareNetworkAge

CompareNetworkAge

Compare the network age to a number

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetComparison(comparisonType, value)

IronPython.Runtime.PythonTuple GetComparison()

Delete()

Properties

Position

Name

HdkApi.RepertoireEditor.Conditions.ComparePreviousBehaviour

ComparePreviousBehaviour

Compare the previous behaviour to behaviour

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetComparison(comparisonType, behaviour)

IronPython.Runtime.PythonTuple GetComparison()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.ComparePreviousState

ComparePreviousState

Compare the previous state to a state

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetComparison(comparisonType, state)

IronPython.Runtime.PythonTuple GetComparison()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.CompareRegister

CompareRegister

Compare a register to a value

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetComparison(type, register, value)

IronPython.Runtime.PythonTuple GetComparison()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.CompareStateAge

CompareStateAge

Compare the current state age to a value

Namespace: [HdkApi.RepertoireEditor.Conditions](#)
Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetComparison(comparisonType, value)

IronPython.Runtime.PythonTuple GetComparison()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.ComparisonType

ComparisonType

An enumeration of comparison types

Namespace: [HdkApi.RepertoireEditor.Conditions](#)
Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Fields

Equal

Not_Equal

Less_Than

Less_Than_Or_Equal_To

Greater_Than

Greater_Than_Or_Equal_To

HdkApi.RepertoireEditor.Conditions.Condition

Condition

Define condition logic for use with behaviours, actions and states

Namespace: [HdkApi.RepertoireEditor.Conditions](#)
Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

HdkApi.RepertoireEditor.Conditions.CompareCurrentBehaviour CreateCompareCurrentBehaviour()

HdkApi.RepertoireEditor.Conditions.CompareCurrentNetworkAge CreateCompareCurrentNetworkAge()

HdkApi.RepertoireEditor.Conditions.CompareNetworkAge CreateCompareNetworkAge()

HdkApi.RepertoireEditor.Conditions.ComparePreviousBehaviour CreateComparePreviousBehaviour()

HdkApi.RepertoireEditor.Conditions.ComparePreviousState CreateComparePreviousState()

HdkApi.RepertoireEditor.Conditions.CompareRegister CreateCompareRegister()

HdkApi.RepertoireEditor.Conditions.CompareStateAge CreateCompareStateAge()

HdkApi.RepertoireEditor.Conditions.And CreateAnd()

HdkApi.RepertoireEditor.Conditions.Or CreateOr()

HdkApi.RepertoireEditor.Conditions.RandomValue CreateRandomValue()

System.Object GetInput()

Properties

Position

Name

HdkApi.RepertoireEditor.Conditions.Or

Or

Performs an or operation on inputs

Namespace: [HdkApi.RepertoireEditor.Conditions](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

System.Collections.Generic.List[System.Object] GetInputs()

HdkApi.RepertoireEditor.Conditions.CompareCurrentBehaviour CreateCompareCurrentBehaviour()

HdkApi.RepertoireEditor.Conditions.CompareCurrentNetworkAge CreateCompareCurrentNetworkAge()

HdkApi.RepertoireEditor.Conditions.CompareNetworkAge CreateCompareNetworkAge()

HdkApi.RepertoireEditor.Conditions.ComparePreviousBehaviour CreateComparePreviousBehaviour()

HdkApi.RepertoireEditor.Conditions.ComparePreviousState CreateComparePreviousState()

HdkApi.RepertoireEditor.Conditions.CompareRegister CreateCompareRegister()

HdkApi.RepertoireEditor.Conditions.CompareStateAge CreateCompareStateAge()

HdkApi.RepertoireEditor.Conditions.And CreateAnd()

HdkApi.RepertoireEditor.Conditions.Or CreateOr()

HdkApi.RepertoireEditor.Conditions.RandomValue CreateRandomValue()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Conditions.RandomValue

RandomValue

A random value from a selection or range

Namespace: [HdkApi.RepertoireEditor.Conditions](#)
Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

- SetComparison(type, value, range)
- SetComparison(type, value, intSelection)
- IronPython.Runtime.PythonTuple GetComparison()
- Delete()

Properties

- Position
- Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
```

HdkApi.RepertoireEditor.Emotes

Summary

Type
Description

Assembly
ActionEmote
An emote to trigger an action
Home.Editors.RepertoireEditor.Scripting
BehaviourEmote
An emote to trigger a behaviour
Home.Editors.RepertoireEditor.Scripting
EmoteContext
Group emotes into sub-menus
Home.Editors.RepertoireEditor.Scripting
EmoteGroup
Group emotes into sub-menus
Home.Editors.RepertoireEditor.Scripting
GlobalEmotes
Global emotes are emotes that are available in every region.
Home.Editors.RepertoireEditor.Scripting
RegionalEmotes
Determines which region these emotes will be used
Home.Editors.RepertoireEditor.Scripting

HdkApi.RepertoireEditor.Emotes.ActionEmote

ActionEmote

An emote to trigger an action

Namespace: [HdkApi.RepertoireEditor.Emotes](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetAction(action)

HdkApi.RepertoireEditor.Action GetAction()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateEmotes():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        print("create a test behaviour and a test action")
        behaviour = rep.CreateBehaviour("Behaviour")
        action = rep.CreateAction("Action")

        if(behaviour and action):
            behaviour.Position = Point(500,100)
            action.Position = Point(500,100)

            globalEmotes = rep.CreateGlobalEmotes("global")
            print("Created global emotes")
            if(globalEmotes):
                globalEmotes.Position = Point(500,100)

            regionalSitIdleContext = globalEmotes.SitIdle
            regionalSitOnFloorContext = globalEmotes.SitOnFloor
            regionalStandIdleContext = globalEmotes.StandIdle

            if(regionalSitIdleContext):
                someGroup = regionalSitOnFloorContext.CreateEmoteGroup("group")
                print("Created an emote group within the sit on floor context")
                if(someGroup):
                    someGroup.Position = Point(500,100)

                    actionEmote = someGroup.CreateActionEmote("action_emote")
                    print("Created an action emote")
                    if(actionEmote):
                        actionEmote.Position = Point(500,100)
                        actionEmote.SetAction(action)

                    behaviourEmote = someGroup.CreateBehaviourEmote("behaviour_emote")
                    print("Created a behaviour emote")
                    if(behaviourEmote):
                        behaviourEmote.Position = Point(500,200)
                        behaviourEmote.SetBehaviour(behaviour)

    CreateEmotes()
```

HdkApi.RepertoireEditor.Emotes.BehaviourEmote

BehaviourEmote

An emote to trigger a behaviour

Namespace: [HdkApi.RepertoireEditor.Emotes](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetBehaviour(behaviour)

HdkApi.RepertoireEditor.Behaviour GetBehaviour()

Delete()

Properties

Position

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateEmotes():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        print("create a test behaviour and a test action")
        behaviour = rep.CreateBehaviour("Behaviour")
        action = rep.CreateAction("Action")

        if(behaviour and action):
            behaviour.Position = Point(500,100)
            action.Position = Point(500,100)

            globalEmotes = rep.CreateGlobalEmotes("global")
            print("Created global emotes")
            if(globalEmotes):
                globalEmotes.Position = Point(500,100)

            regionalSitIdleContext = globalEmotes.SitIdle
            regionalSitOnFloorContext = globalEmotes.SitOnFloor
            regionalStandIdleContext = globalEmotes.StandIdle

            if(regionalSitIdleContext):
                someGroup = regionalSitOnFloorContext.CreateEmoteGroup("group")
                print("Created an emote group within the sit on floor context")
                if(someGroup):
                    someGroup.Position = Point(500,100)

                    actionEmote = someGroup.CreateActionEmote("action_emote")
                    print("Created an action emote")
                    if(actionEmote):
                        actionEmote.Position = Point(500,100)
                        actionEmote.SetAction(action)

                    behaviourEmote = someGroup.CreateBehaviourEmote("behaviour_emote")
                    print("Created a behaviour emote")
                    if(behaviourEmote):
                        behaviourEmote.Position = Point(500,200)
                        behaviourEmote.SetBehaviour(behaviour)

    CreateEmotes()

```

HdkApi.RepertoireEditor.Emotes.EmoteContext

EmoteContext

Group emotes into sub-menus

Namespace: [HdkApi.RepertoireEditor.Emotes](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Remarks

 The available emote contexts are specified by the client and may not be created, deleted or renamed.

Methods

HdkApi.RepertoireEditor.Emotes.EmoteGroup CreateEmoteGroup(groupName)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Emotes.EmoteGroup] GetEmoteGroups()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateEmotes():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        print("create a test behaviour and a test action")
        behaviour = rep.CreateBehaviour("Behaviour")
        action = rep.CreateAction("Action")

        if(behaviour and action):
            behaviour.Position = Point(500,100)
            action.Position = Point(500,100)

            globalEmotes = rep.CreateGlobalEmotes("global")
            print("Created global emotes")
            if(globalEmotes):
                globalEmotes.Position = Point(500,100)

            regionalSitIdleContext = globalEmotes.SitIdle
            regionalSitOnFloorContext = globalEmotes.SitOnFloor
            regionalStandIdleContext = globalEmotes.StandIdle

            if(regionalSitIdleContext):
                someGroup = regionalSitOnFloorContext.CreateEmoteGroup("group")
                print("Created an emote group within the sit on floor context")
                if(someGroup):
                    someGroup.Position = Point(500,100)

                    actionEmote = someGroup.CreateActionEmote("action_emote")
                    print("Created an action emote")
                    if(actionEmote):
                        actionEmote.Position = Point(500,100)
                        actionEmote.SetAction(action)

                    behaviourEmote = someGroup.CreateBehaviourEmote("behaviour_emote")
                    print("Created a behaviour emote")
                    if(behaviourEmote):
                        behaviourEmote.Position = Point(500,200)
                        behaviourEmote.SetBehaviour(behaviour)

    CreateEmotes()
```

HdkApi.RepertoireEditor.Emotes.EmoteGroup

EmoteGroup

Group emotes into sub-menus

Namespace: [HdkApi.RepertoireEditor.Emotes](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

System.Collections.Generic.List[HdkApi.RepertoireEditor.Emotes.ActionEmote] GetActionEmotes()

HdkApi.RepertoireEditor.Emotes.ActionEmote CreateActionEmote(emoteName)

System.Collections.Generic.List[HdkApi.RepertoireEditor.Emotes.BehaviourEmote] GetBehaviourEmotes()

HdkApi.RepertoireEditor.Emotes.BehaviourEmote CreateBehaviourEmote(emoteName)

Delete()

Properties

Position

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateEmotes():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        print("create a test behaviour and a test action")
        behaviour = rep.CreateBehaviour("Behaviour")
        action = rep.CreateAction("Action")

        if(behaviour and action):
            behaviour.Position = Point(500,100)
            action.Position = Point(500,100)

            globalEmotes = rep.CreateGlobalEmotes("global")
            print("Created global emotes")
            if(globalEmotes):
                globalEmotes.Position = Point(500,100)

            regionalSitIdleContext = globalEmotes.SitIdle
            regionalSitOnFloorContext = globalEmotes.SitOnFloor
            regionalStandIdleContext = globalEmotes.StandIdle

            if(regionalSitIdleContext):
                someGroup = regionalSitOnFloorContext.CreateEmoteGroup("group")
                print("Created an emote group within the sit on floor context")
                if(someGroup):
                    someGroup.Position = Point(500,100)

                    actionEmote = someGroup.CreateActionEmote("action_emote")
                    print("Created an action emote")
                    if(actionEmote):
                        actionEmote.Position = Point(500,100)
                        actionEmote.SetAction(action)

                    behaviourEmote = someGroup.CreateBehaviourEmote("behaviour_emote")
                    print("Created a behaviour emote")
                    if(behaviourEmote):
                        behaviourEmote.Position = Point(500,200)
                        behaviourEmote.SetBehaviour(behaviour)

    CreateEmotes()

```

HdkApi.RepertoireEditor.Emotes.GlobalEmotes

GlobalEmotes

Global emotes are emotes that are available in every region.

Namespace: [HdkApi.RepertoireEditor.Emotes](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

Properties

StandIdle
SitIdle
SitOnFloor
Position
Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateEmotes():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        print("create a test behaviour and a test action")
        behaviour = rep.CreateBehaviour("Behaviour")
        action = rep.CreateAction("Action")

        if(behaviour and action):
            behaviour.Position = Point(500,100)
            action.Position = Point(500,100)

            globalEmotes = rep.CreateGlobalEmotes("global")
            print("Created global emotes")
            if(globalEmotes):
                globalEmotes.Position = Point(500,100)

            regionalSitIdleContext = globalEmotes.SitIdle
            regionalSitOnFloorContext = globalEmotes.SitOnFloor
            regionalStandIdleContext = globalEmotes.StandIdle

            if(regionalSitIdleContext):
                someGroup = regionalSitOnFloorContext.CreateEmoteGroup("group")
                print("Created an emote group within the sit on floor context")
                if(someGroup):
                    someGroup.Position = Point(500,100)

                    actionEmote = someGroup.CreateActionEmote("action_emote")
                    print("Created an action emote")
                    if(actionEmote):
                        actionEmote.Position = Point(500,100)
                        actionEmote.SetAction(action)

                    behaviourEmote = someGroup.CreateBehaviourEmote("behaviour_emote")
                    print("Created a behaviour emote")
                    if(behaviourEmote):
                        behaviourEmote.Position = Point(500,200)
                        behaviourEmote.SetBehaviour(behaviour)

    CreateEmotes()
```

HdkApi.RepertoireEditor.Emotes.RegionalEmotes

RegionalEmotes

Determines which region these emotes will be used

Namespace: [HdkApi.RepertoireEditor.Emotes](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

Delete()

Properties

Region

StandIdle

SitIdle

SitOnFloor

Position

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
def CreateEmotes():
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        print("create a test behaviour and a test action")
        behaviour = rep.CreateBehaviour("Behaviour")
        action = rep.CreateAction("Action")

        if(behaviour and action):
            behaviour.Position = Point(500,100)
            action.Position = Point(500,100)

            regionalEmotes = rep.CreateRegionalEmotes("japan")
            print("Created regional emotes")
            if(regionalEmotes):
                regionalEmotes.Position = Point(500,100)
                regionalEmotes.Region = "ja-JP"

                regionalSitIdleContext = regionalEmotes.SitIdle
                regionalSitOnFloorContext = regionalEmotes.SitOnFloor
                regionalStandIdleContext = regionalEmotes.StandIdle

                if(regionalSitIdleContext):
                    someGroup = regionalSitIdleContext.CreateEmoteGroup("group")
                    print("Created an emote group within the sit idle context")
                    if(someGroup):
                        someGroup.Position = Point(500,100)

                        actionEmote = someGroup.CreateActionEmote("action_emote")
                        print("Created an action emote")
                        if(actionEmote):
                            actionEmote.Position = Point(500,100)
                            actionEmote.SetAction(action)

                        behaviourEmote = someGroup.CreateBehaviourEmote("behaviour_emote")
                        print("Created a behaviour emote")
                        if(behaviourEmote):
                            behaviourEmote.Position = Point(500,200)
                            behaviourEmote.SetBehaviour(behaviour)

    CreateEmotes()

```

HdkApi.RepertoireEditor.Outputs

Summary

Type
Description
Assembly
ClearRegister

Clear a register value as part of an output
Home.Editors.RepertoireEditor.Scripting
OutputType
Whether onEntry or onExit
Home.Editors.RepertoireEditor.Scripting
Output
Outputs are a way of defining behaviour before or after an action/behaviour/state is triggered
Home.Editors.RepertoireEditor.Scripting
SetNextBehaviour
Set the next behaviour as part of an output
Home.Editors.RepertoireEditor.Scripting
SetRegister
Set a register value as part of an output
Home.Editors.RepertoireEditor.Scripting
SpawnAction
Spawn an action as part of an output
Home.Editors.RepertoireEditor.Scripting
TriggerEvent
Trigger an event as part of an output
Home.Editors.RepertoireEditor.Scripting

HdkApi.RepertoireEditor.Outputs.ClearRegister

ClearRegister

Clear a register value as part of an output

Namespace: [HdkApi.RepertoireEditor.Outputs](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetRegister(register)

HdkApi.RepertoireEditor.RepertoireRegister GetRegister()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateOutput():
    repertoire = RepertoireEditor.CurrentRepertoire

    #create action
    action = repertoire.CreateAction("action")

    #add an output
    output = action.CreateOutput("output")

    #add a few output types
    clearRegister = output.CreateClearRegister("clearRegister", OutputType.OnEntry)
    clearRegister.SetRegister(register)

    setNextBehaviour = output.CreateSetNextBehaviour("setNextBehaviour", OutputType.OnEntry)
    setNextBehaviour.SetBehaviour(behaviour)

    setRegister = output.CreateSetRegister("setRegister", OutputType.OnEntry)
    setRegister.SetValue(register, value)

    spawnAction = output.CreateSpawnAction("spawnAction", OutputType.OnExit)
    spawnAction.SetAction(action)

    triggerEvent = output.CreateTriggerEvent("triggerEvent", OutputType.OnExit)
    triggerEvent.SetEvent(event)

    CreateOutput()
```

HdkApi.RepertoireEditor.Outputs.Output

Output

Outputs are a way of defining behaviour before or after an action/behaviour/state is triggered

Namespace: [HdkApi.RepertoireEditor.Outputs](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

System.Collections.Generic.List[HdkApi.RepertoireEditor.Outputs.BaseOutput] GetOutputs(type)

HdkApi.RepertoireEditor.Outputs.ClearRegister CreateClearRegister(type)
HdkApi.RepertoireEditor.Outputs.SetNextBehaviour CreateSetNextBehaviour(type)
HdkApi.RepertoireEditor.Outputs.SetRegister CreateSetRegister(type)
HdkApi.RepertoireEditor.Outputs.SpawnAction CreateSpawnAction(type)
HdkApi.RepertoireEditor.Outputs.TriggerEvent CreateTriggerEvent(type)
Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateOutput():
    repertoire = RepertoireEditor.CurrentRepertoire

    #create action
    action = repertoire.CreateAction("action")

    #add an output
    output = action.CreateOutput("output")

    #add a few output types
    clearRegister = output.CreateClearRegister("clearRegister", OutputType.OnEntry)
    clearRegister.SetRegister(register)

    setNextBehaviour = output.CreateSetNextBehaviour("setNextBehaviour", OutputType.OnEntry)
    setNextBehaviour.SetBehaviour(behaviour)

    setRegister = output.CreateSetRegister("setRegister", OutputType.OnEntry)
    setRegister.SetValue(register, value)

    spawnAction = output.CreateSpawnAction("spawnAction", OutputType.OnExit)
    spawnAction.SetAction(action)

    triggerEvent = output.CreateTriggerEvent("triggerEvent", OutputType.OnExit)
    triggerEvent.SetEvent(event)

    CreateOutput()
```

HdkApi.RepertoireEditor.Outputs.OutputType

OutputType

Whether onEntry or onExit

Namespace: [HdkApi.RepertoireEditor.Outputs](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Fields

OnEntry

OnExit

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateOutput():
    repertoire = RepertoireEditor.CurrentRepertoire

    #create action
    action = repertoire.CreateAction("action")

    #add an output
    output = action.CreateOutput("output")

    #add a few output types
    clearRegister = output.CreateClearRegister("clearRegister", OutputType.OnEntry)
    clearRegister.SetRegister(register)

    setNextBehaviour = output.CreateSetNextBehaviour("setNextBehaviour", OutputType.OnEntry)
    setNextBehaviour.SetBehaviour(behaviour)

    setRegister = output.CreateSetRegister("setRegister", OutputType.OnEntry)
    setRegister.SetValue(register, value)

    spawnAction = output.CreateSpawnAction("spawnAction", OutputType.OnExit)
    spawnAction.SetAction(action)

    triggerEvent = output.CreateTriggerEvent("triggerEvent", OutputType.OnExit)
    triggerEvent.SetEvent(event)

    CreateOutput()
```

HdkApi.RepertoireEditor.Outputs.SetNextBehaviour

SetNextBehaviour

Set the next behaviour as part of an output

Namespace: [HdkApi.RepertoireEditor.Outputs](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Behaviour GetBehaviour()

SetBehaviour(behaviour)

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateOutput():
    repertoire = RepertoireEditor.CurrentRepertoire

    #create action
    action = repertoire.CreateAction("action")

    #add an output
    output = action.CreateOutput("output")

    #add a few output types
    clearRegister = output.CreateClearRegister("clearRegister", OutputType.OnEntry)
    clearRegister.SetRegister(register)

    setNextBehaviour = output.CreateSetNextBehaviour("setNextBehaviour", OutputType.OnEntry)
    setNextBehaviour.SetBehaviour(behaviour)

    setRegister = output.CreateSetRegister("setRegister", OutputType.OnEntry)
    setRegister.SetValue(register, value)

    spawnAction = output.CreateSpawnAction("spawnAction", OutputType.OnExit)
    spawnAction.SetAction(action)

    triggerEvent = output.CreateTriggerEvent("triggerEvent", OutputType.OnExit)
    triggerEvent.SetEvent(event)

CreateOutput()
```

HdkApi.RepertoireEditor.Outputs.SetRegister

SetRegister

Set a register value as part of an output

Namespace: [HdkApi.RepertoireEditor.Outputs](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetValue(register, value)

IronPython.Runtime.PythonTuple GetValue()

Delete()

Properties

Position

Name

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateOutput():
    repertoire = RepertoireEditor.CurrentRepertoire

    #create action
    action = repertoire.CreateAction("action")

    #add an output
    output = action.CreateOutput("output")

    #add a few output types
    clearRegister = output.CreateClearRegister("clearRegister", OutputType.OnEntry)
    clearRegister.SetRegister(register)

    setNextBehaviour = output.CreateSetNextBehaviour("setNextBehaviour", OutputType.OnEntry)
    setNextBehaviour.SetBehaviour(behaviour)

    setRegister = output.CreateSetRegister("setRegister", OutputType.OnEntry)
    setRegister.SetValue(register, value)

    spawnAction = output.CreateSpawnAction("spawnAction", OutputType.OnExit)
    spawnAction.SetAction(action)

    triggerEvent = output.CreateTriggerEvent("triggerEvent", OutputType.OnExit)
    triggerEvent.SetEvent(event)

    CreateOutput()
```

HdkApi.RepertoireEditor.Outputs.SpawnAction

SpawnAction

Spawn an action as part of an output

Namespace: [HdkApi.RepertoireEditor.Outputs](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.Action GetAction()

SetAction(action)

Delete()

Properties

Position

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateOutput():
    repertoire = RepertoireEditor.CurrentRepertoire

    #create action
    action = repertoire.CreateAction("action")

    #add an output
    output = action.CreateOutput("output")

    #add a few output types
    clearRegister = output.CreateClearRegister("clearRegister", OutputType.OnEntry)
    clearRegister.SetRegister(register)

    setNextBehaviour = output.CreateSetNextBehaviour("setNextBehaviour", OutputType.OnEntry)
    setNextBehaviour.SetBehaviour(behaviour)

    setRegister = output.CreateSetRegister("setRegister", OutputType.OnEntry)
    setRegister.SetValue(register, value)

    spawnAction = output.CreateSpawnAction("spawnAction", OutputType.OnExit)
    spawnAction.SetAction(action)

    triggerEvent = output.CreateTriggerEvent("triggerEvent", OutputType.OnExit)
    triggerEvent.SetEvent(event)

CreateOutput()

```

HdkApi.RepertoireEditor.Outputs.TriggerEvent

TriggerEvent

Trigger an event as part of an output

Namespace: [HdkApi.RepertoireEditor.Outputs](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

SetEvent(repertoireEvent, value)

IronPython.Runtime.PythonTuple GetEvent()

Delete()

Properties

Position

Name

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

def CreateOutput():
    repertoire = RepertoireEditor.CurrentRepertoire

    #create action
    action = repertoire.CreateAction("action")

    #add an output
    output = action.CreateOutput("output")

    #add a few output types
    clearRegister = output.CreateClearRegister("clearRegister", OutputType.OnEntry)
    clearRegister.SetRegister(register)

    setNextBehaviour = output.CreateSetNextBehaviour("setNextBehaviour", OutputType.OnEntry)
    setNextBehaviour.SetBehaviour(behaviour)

    setRegister = output.CreateSetRegister("setRegister", OutputType.OnEntry)
    setRegister.SetValue(register, value)

    spawnAction = output.CreateSpawnAction("spawnAction", OutputType.OnExit)
    spawnAction.SetAction(action)

    triggerEvent = output.CreateTriggerEvent("triggerEvent", OutputType.OnExit)
    triggerEvent.SetEvent(event)

CreateOutput()

```

HdkApi.RepertoireEditor.UserInterface

Summary

Type
Description
Assembly
Menu
Menu represents a menu in the Scene Editor UI, new items can be added to the menu by calling the AddMenuItem function, a new Menu can be created through MenuManager.CreateMenu
Home.Editors.RepertoireEditor.Scripting
MenuItem
MenuItem represents an item within a Menu in the Scene Editor UI, it provides access to properties about the menu item such as what icon it uses and whether it has a button associated with it, for a menu item to be visible it must be added to a Menu through AddMenuItem, to create a new MenuItem call MenuManager.CreateMenuItem

Home.Editors.RepertoireEditor.Scripting

MenuManager

MenuManager is used to add new menus and menu items to the Scene Editor UI, to add a new menu call CreateMenu, to add a new menu item call CreateMenuItem and then pass the returned MenuItem to a menu's AddMenuItem function

Home.Editors.RepertoireEditor.Scripting

HdkApi.RepertoireEditor.UserInterface.Menu

Menu

Menu represents a menu in the Scene Editor UI, new items can be added to the menu by calling the AddMenuItem function, a new Menu can be created through MenuManager.CreateMenu

Namespace: [HdkApi.RepertoireEditor.UserInterface](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

AddMenuItem(menuItem)

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("My Hdk API Scripts")

        actionMenuItem = self.createActionMenu()
        behaviourMenuItem = self.createBehaviourMenu()

        # Add the MenuItem to the example menu
        menu.AddMenuItem(actionMenuItem)
        menu.AddMenuItem(behaviourMenuItem)

    def createActionMenu(self):
        # Create a new MenuItem
        itemDisplayText = "CreateAction"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Create an action with one state"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.A
        #menuItem.HasToolBarButton = True

        # Hook up the execute event to set what happens when the menu item is pressed
```

```

menuItem.Execute += self.createAction

# Hook up the can execute event to set whether the menu item is enabled
menuItem.CanExecute += self.repertoireCanExecute

return menuItem

def createBehaviourMenu(self):
    # Create a new MenuItem
    itemDisplayText = "CreateBehaviour"
    menuItem = MenuManager.CreateMenuItem(itemDisplayText)
    menuItem.Description = "Create a behaviour with one state"
    menuItem.Toggleable = True
    menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.B
    #menuItem.HasToolBarButton = True

    # Hook up the execute event to set what happens when the menu item is pressed
    menuItem.Execute += self.createBehaviour

    # Hook up the can execute event to set whether the menu item is enabled
    menuItem.CanExecute += self.repertoireCanExecute

    return menuItem

def repertoireCanExecute(self):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep is None):
        return False
    return True

def createBehaviour(self, sender, args):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        behaviour = rep.CreateBehaviour("behaviour")
        if(behaviour):
            behaviour.Position = Point(500,100)

            state = behaviour.CreateState("state")
            if(state):
                state.Position = Point(500,100)

    print("Created behaviour")

def createAction(self, sender, args):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        action = rep.CreateAction("action")
        action.Position = Point(416,192)

        state = action.CreateState("state")
        if(state):
            state.Position = Point(500,100)

```

```
print("Created action")

example = UiExample()
```

Associated Types

- [HdkApi.RepertoireEditor.UserInterface.MenuManager](#)
- [HdkApi.RepertoireEditor.UserInterface.Menuitem](#)

HdkApi.RepertoireEditor.UserInterface.Menuitem

Menuitem

Menuitem represents an item within a Menu in the Scene Editor UI, it provides access to properties about the menu item such as what icon it uses and whether it has a button associated with it, for a menu item to be visible it must be added to a Menu through AddMenuitem, to create a new Menuitem call MenuManager.CreateMenuitem

Namespace: [HdkApi.RepertoireEditor.UserInterface](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Properties

DisplayText

Description

IconPath

Shortcut

Toggleable

HasToolbarButton

Events

Execute

CanExecute

Delegates

CanExecuteHandler

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
```

```

# Create a new menu
menu = MenuManager.CreateMenu("My Hdk API Scripts")

actionMenuItem = self.createActionMenu()
behaviourMenuItem = self.createBehaviourMenu()

# Add the MenuItem to the example menu
menu.AddMenuItem(actionMenuItem)
menu.AddMenuItem(behaviourMenuItem)

def createActionMenu(self):
    # Create a new MenuItem
    itemDisplayText = "CreateAction"
    menuItem = MenuManager.CreateMenuItem(itemDisplayText)
    menuItem.Description = "Create an action with one state"
    menuItem.Toggleable = True
    menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.A
    #menuItem.HasToolBarButton = True

    # Hook up the execute event to set what happens when the menu item is pressed
    menuItem.Execute += self.createAction

    # Hook up the can execute event to set whether the menu item is enabled
    menuItem.CanExecute += self.repertoireCanExecute

    return menuItem

def createBehaviourMenu(self):
    # Create a new MenuItem
    itemDisplayText = "CreateBehaviour"
    menuItem = MenuManager.CreateMenuItem(itemDisplayText)
    menuItem.Description = "Create a behaviour with one state"
    menuItem.Toggleable = True
    menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.B
    #menuItem.HasToolBarButton = True

    # Hook up the execute event to set what happens when the menu item is pressed
    menuItem.Execute += self.createBehaviour

    # Hook up the can execute event to set whether the menu item is enabled
    menuItem.CanExecute += self.repertoireCanExecute

    return menuItem

def repertoireCanExecute(self):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep is None):
        return False
    return True

def createBehaviour(self, sender, args):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        behaviour = rep.CreateBehaviour("behaviour")
        if(behaviour):
            behaviour.Position = Point(500,100)

            state = behaviour.CreateState("state")
            if(state):
                state.Position = Point(500,100)

        print("Created behaviour")

def createAction(self, sender, args):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        action = rep.CreateAction("action")
        action.Position = Point(416,192)

```

```
state = action.CreateState("state")
if(state):
    state.Position = Point(500,100)
```

```
print("Created action")

example = UiExample()
```

Associated Types

- [HdkApi.RepertoireEditor.UserInterface.MenuManager](#)
- [HdkApi.RepertoireEditor.UserInterface.Menu](#)

HdkApi.RepertoireEditor.UserInterface.MenuManager

MenuManager

MenuManager is used to add new menus and menu items to the Scene Editor UI, to add a new menu call CreateMenu, to add a new menu item call CreateMenuItem and then pass the returned MenuItem to a menu's AddMenuItem function

Namespace: [HdkApi.RepertoireEditor.UserInterface](#)

Assembly: *Home.Editors.RepertoireEditor.Scripting.dll*

Methods

HdkApi.RepertoireEditor.UserInterface.Menu MenuManager.CreateMenu(name)

HdkApi.RepertoireEditor.UserInterface.Menu MenuManager.GetMenu(name)

HdkApi.RepertoireEditor.UserInterface.MenuItem MenuManager.CreateMenuItem(displayText)

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.65.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>
# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("My Hdk API Scripts")

        actionMenuItem = self.createActionMenu()
        behaviourMenuItem = self.createBehaviourMenu()

        # Add the MenuItem to the example menu
        menu.AddMenuItem(actionMenuItem)
        menu.AddMenuItem(behaviourMenuItem)

    def createActionMenu(self):
        # Create a new MenuItem
        itemDisplayText = "CreateAction"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Create an action with one state"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.A
```

```

#menuItem.HasToolBarButton = True

# Hook up the execute event to set what happens when the menu item is pressed
menuItem.Execute += self.createAction

# Hook up the can execute event to set whether the menu item is enabled
menuItem.CanExecute += self.repertoireCanExecute

return menuItem

def createBehaviourMenu(self):
    # Create a new MenuItem
    itemDisplayText = "CreateBehaviour"
    menuItem = MenuManager.CreateMenuItem(itemDisplayText)
    menuItem.Description = "Create a behaviour with one state"
    menuItem.Toggleable = True
    menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.B
    #menuItem.HasToolBarButton = True

    # Hook up the execute event to set what happens when the menu item is pressed
    menuItem.Execute += self.createBehaviour

    # Hook up the can execute event to set whether the menu item is enabled
    menuItem.CanExecute += self.repertoireCanExecute

    return menuItem

def repertoireCanExecute(self):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep is None):
        return False
    return True

def createBehaviour(self, sender, args):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        behaviour = rep.CreateBehaviour("behaviour")
        if(behaviour):
            behaviour.Position = Point(500,100)

            state = behaviour.CreateState("state")
            if(state):
                state.Position = Point(500,100)

    print("Created behaviour")

def createAction(self, sender, args):
    rep = RepertoireEditor.CurrentRepertoire
    if(rep):
        action = rep.CreateAction("action")
        action.Position = Point(416,192)

        state = action.CreateState("state")
        if(state):
            state.Position = Point(500,100)

```

```
        print("Created action")

example = UiExample()
```

Associated Types

- [HdkApi.RepertoireEditor.UserInterface.Menu](#)
- [HdkApi.RepertoireEditor.UserInterface.Menuitem](#)

HdkApi.SceneEditor

Summary

Type
Description
Assembly
SceneEditor
Provides access to core functionality of the Scene Editor, such as getting the currently open scene.
Home.Editors.SceneEditor.Scripting

HdkApi.SceneEditor.SceneEditor

SceneEditor

Provides access to core functionality of the Scene Editor, such as getting the currently open scene.

Namespace: [HdkApi.SceneEditor](#)

Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Methods

SceneEditor.OpenSceneProject(scprojPath)

System.Boolean SceneEditor.Export(validate)

Properties

CurrentScene

Events

OnSaving

OnSaved

OnOpened

OnExporting

OnExported

OnPackaging

OnPackaged

Delegates

SceneAction

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the SceneEditor class
from HdkApi.SceneEditor import SceneEditor

# Import the HomeScene class
from HdkApi.Data.Scene import HomeScene

class SceneEditorExample():
    def __init__(self):
        # Get the current scene which is open in the Scene Editor
        openScene = SceneEditor.CurrentScene

        # Print the scene's filepath
        print openScene.FilePath

        # Export the currently open scene to a (.scene) file so it can be tested in game
        validate = True
        succeeded = SceneEditor.Export(validate)
        print succeeded

        self.RegisterCallbacks()

    def RegisterCallbacks(self):
        # Hook up to the OnOpened event called when a scene is opened in the Scene Editor
        SceneEditor.OnOpened += self.onOpened

        # Hook up to the OnSaving event called when a scene is about to be saved
        SceneEditor.OnSaving += self.onSaving

        # Hook up to the OnSaved event called when a scene has been successfully saved
        SceneEditor.OnSaved += self.onSaved

        # Hook up to the OnExporting event called when a scene is about to be exported from an .scproj
        # file to a .scene file
        SceneEditor.OnExporting += self.onExporting

        # Hook up to the OnExported event called when a scene has been successfully exported from an
        # .scproj file to a .scene file
        SceneEditor.OnExported += self.onExported

        # Hook up to the OnPackaging event called when a scene is about to be packaged
        SceneEditor.OnPackaging += self.onPackaging

        # Hook up to the OnPackaged event called when a scene has been successfully packaged
        SceneEditor.OnPackaged += self.onPackaged

        # Example of a callback function triggered when a scene is opened in the Scene Editor
```

```
def onOpened(self, scene):
    print "Opened: {0}".format(scene.FilePath)

    # Example of a callback function triggered when the scene open in the Scene Editor is about to be
    saved
    def onSaving(self, scene):
        print "Saving: {0}".format(scene.FilePath)

    # Example of a callback function triggered when the scene open in the Scene Editor has been
    successfully saved
    def onSaved(self, scene):
        print "Saved: {0}".format(scene.FilePath)

    # Example of a callback function triggered when the scene open in the Scene Editor is about to be
    exported
    def onExporting(self, scene):
        print "Exporting: {0}".format(scene.FilePath)

    # Example of a callback function triggered when the scene open in the Scene Editor has been
    successfully exported
    def onExported(self, scene):
        print "Exported: {0}".format(scene.FilePath)

    # Example of a callback function triggered when the scene open in the Scene Editor is about to be
    packaged
    def onPackaging(self, scene):
        print "Packaging: {0}".format(scene.FilePath)

    # Example of a callback function triggered when the scene open in the Scene Editor is about to be
```

```
packaged
def onPackaged(self, scene):
    print "Packaged: {0}".format(scene.FilePath)
```

Associated Types

- [HdkApi.Data.Scene.HomeScene](#)

HdkApi.SceneEditor.Camera

Summary

Type
Description
Assembly
Camera
The Camera class is the basis for derived camera classes in the HdkApi such as SceneCamera.
Home.Editors.SceneEditor.Scripting
CameraManager
The CameraManager has access to Camera objects from the SceneEditor. Currently it provides access to the perspective camera that is used to navigate in the scene editor.
Home.Editors.SceneEditor.Scripting
SceneCamera
The SceneCamera class represents the perspective camera that is used to navigate the scene in the editor. You may use this class to retrieve position and look at point data.
Home.Editors.SceneEditor.Scripting

HdkApi.SceneEditor.Camera.Camera

Camera

The Camera class is the basis for derived camera classes in the HdkApi such as SceneCamera.

Namespace: [HdkApi.SceneEditor.Camera](#)

Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Properties

Position

LookAt

Events

CameraChanged

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Camera classes
from HdkApi.SceneEditor.Camera import CameraManager, SceneCamera

class CameraExample():
    def __init__(self):
        self.camera = CameraManager.SceneCamera
        self.camera.CameraChanged += self.cameraUpdated

    def printPosition(self):
        str = "{0}{1}{2}".format(self.camera.Position.X, self.camera.Position.Y, self.camera.Position.Z)
        print(str)

    def printLookAt(self):
        str = "{0}{1}{2}".format(self.camera.LookAt.X, self.camera.LookAt.Y, self.camera.LookAt.Z)
        print(str)

    def cameraUpdated(self, sender, args):
        print("Camera Updated")
        self.printPosition()
        self.printLookAt()

camExample = CameraExample()
```

Associated Types

- [HdkApi.SceneEditor.Camera.CameraManager](#)
- [HdkApi.SceneEditor.Camera.SceneCamera](#)

HdkApi.SceneEditor.Camera.CameraManager

CameraManager

The CameraManager has access to Camera objects from the SceneEditor. Currently it provides access to the perspective camera that is used to navigate in the scene editor.

Namespace: [HdkApi.SceneEditor.Camera](#)

Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Properties

SceneCamera

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Camera classes
from HdkApi.SceneEditor.Camera import CameraManager, SceneCamera

class CameraExample():
    def __init__(self):
        self.camera = CameraManager.SceneCamera
        self.camera.CameraChanged += self.cameraUpdated

    def printPosition(self):
        str = "{0}{1}{2}".format(self.camera.Position.X, self.camera.Position.Y, self.camera.Position.Z)
        print(str)

    def printLookAt(self):
        str = "{0}{1}{2}".format(self.camera.LookAt.X, self.camera.LookAt.Y, self.camera.LookAt.Z)
        print(str)

    def cameraUpdated(self, sender, args):
        print("Camera Updated")
        self.printPosition()
        self.printLookAt()

camExample = CameraExample()

```

Associated Types

- [HdkApi.SceneEditor.Camera.Camera](#)
- [HdkApi.SceneEditor.Camera.SceneCamera](#)

HdkApi.SceneEditor.Camera.SceneCamera

SceneCamera

The SceneCamera class represents the perspective camera that is used to navigate the scene in the editor. You may use this class to retrieve position and look at point data.

Namespace: [HdkApi.SceneEditor.Camera](#)

Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Methods

[ZoomToSelection\(\)](#)

Properties

[Position](#)

[LookAt](#)

Events

[CameraChanged](#)

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Camera classes
from HdkApi.SceneEditor.Camera import CameraManager, SceneCamera

class CameraExample():
    def __init__(self):
        self.camera = CameraManager.SceneCamera
        self.camera.CameraChanged += self.cameraUpdated

    def printPosition(self):
        str = "{0}{1}{2}".format(self.camera.Position.X, self.camera.Position.Y, self.camera.Position.Z)
        print(str)

    def printLookAt(self):
        str = "{0}{1}{2}".format(self.camera.LookAt.X, self.camera.LookAt.Y, self.camera.LookAt.Z)
        print(str)

    def cameraUpdated(self, sender, args):
        print("Camera Updated")
        self.printPosition()
        self.printLookAt()

camExample = CameraExample()

```

Associated Types

- [HdkApi.SceneEditor.Camera.Camera](#)
- [HdkApi.SceneEditor.Camera.CameraManager](#)

HdkApi.SceneEditor.Selection

Summary

Type
Description
Assembly
SelectionManager
Provides events and data relating to the current selection in the Scene Editor
Home.Editors.SceneEditor.Scripting

HdkApi.SceneEditor.Selection.SelectionManager

SelectionManager

Provides events and data relating to the current selection in the Scene Editor

Namespace: [HdkApi.SceneEditor.Selection](#)

Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Events

SelectionChanged

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the Selection classes
from HdkApi.SceneEditor.Selection import SelectionManager

class SelectionExample():
    def __init__(self):
        # Hook up to the SelectionChanged event
        SelectionManager.SelectionChanged += self.selectionChanged

    def selectionChanged(self, sender, args):
        # Print a message when ever the selection in the Scene Editor changes
        print("Selection Changed")

selExample = SelectionExample()
```

HdkApi.SceneEditor.UserInterface

Summary

Type
Description
Assembly
Menu
Menu represents a menu in the Scene Editor UI, new items can be added to the menu by calling the AddMenuItem function, a new Menu can be created through MenuManager.CreateMenu
Home.Editors.SceneEditor.Scripting
MenuItem
MenuItem represents an item within a Menu in the Scene Editor UI, it provides access to properties about the menu item such as what icon it uses and whether it has a button associated with it, for a menu item to be visible it must be added to a Menu through AddMenuItem, to create a

new MenuItem call MenuManager.CreateMenuItem
Home.Editors.SceneEditor.Scripting
MenuManager
MenuManager is used to add new menus and menu items to the Scene Editor UI, to add a new menu call CreateMenu, to add a new menu item call CreateMenuItem and then pass the returned MenuItem to a menu's AddMenuItem function
Home.Editors.SceneEditor.Scripting

HdkApi.SceneEditor.UserInterface.Menu

Menu

Menu represents a menu in the Scene Editor UI, new items can be added to the menu by calling the AddMenuItem function, a new Menu can be created through MenuManager.CreateMenu

Namespace: [HdkApi.SceneEditor.UserInterface](#)
Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Methods

AddMenuItem(menuItem)

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the UserInterface classes
from HdkApi.SceneEditor.UserInterface import MenuManager, Menu, MenuItem

# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("Example Menu")

        # Create a new MenuItem
        itemDisplayText = "Example Item"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Example menu item used in the UI Example"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.M
        menuItem.HasToolBarButton = True

        # Hook up the execute event to set what happens when the menu item is pressed
        menuItem.Execute += self.exampleExecute

        # Hook up the can execute event to set whether the menu item is enabled
        menuItem.CanExecute += self.exampleCanExecute

        # Add the MenuItem to the example menu
        menu.AddMenuItem(menuItem)

    def exampleCanExecute(self):
        return True

    def exampleExecute(self, sender, args):
        print("Example executed")

example = UiExample()

```

Associated Types

- [HdkApi.SceneEditor.UserInterface.MenuManager](#)
- [HdkApi.SceneEditor.UserInterface.MenuItem](#)

HdkApi.SceneEditor.UserInterface.MenuItem

MenuItem

MenuItem represents an item within a Menu in the Scene Editor UI, it provides access to properties about the menu item such as what icon it uses and whether it has a button associated with it, for a menu item to be visible it must be added to a Menu through AddMenuItem, to create a new MenuItem call MenuManager.CreateMenuItem

Namespace: [HdkApi.SceneEditor.UserInterface](#)
Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Properties

DisplayText

Description

IconPath

Shortcut

Toggleable

HasToolBarButton

Events

Execute

CanExecute

Delegates

CanExecuteHandler

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the UserInterface classes
from HdkApi.SceneEditor.UserInterface import MenuManager, Menu, MenuItem

# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("Example Menu")

        # Create a new MenuItem
        itemDisplayText = "Example Item"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Example menu item used in the UI Example"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.M
        menuItem.HasToolBarButton = True

        # Hook up the execute event to set what happens when the menu item is pressed
        menuItem.Execute += self.exampleExecute

        # Hook up the can execute event to set whether the menu item is enabled
        menuItem.CanExecute += self.exampleCanExecute

        # Add the MenuItem to the example menu
        menu.AddMenuItem(menuItem)

    def exampleCanExecute(self):
        return True

    def exampleExecute(self, sender, args):
        print("Example executed")

example = UiExample()

```

Associated Types

- [HdkApi.SceneEditor.UserInterface.MenuManager](#)
- [HdkApi.SceneEditor.UserInterface.Menu](#)

HdkApi.SceneEditor.UserInterface.MenuManager

MenuManager

MenuManager is used to add new menus and menu items to the Scene Editor UI, to add a new menu call CreateMenu, to add a new menu item call CreateMenuItem and then pass the returned MenuItem to a menu's AddMenuItem function

Namespace: [HdkApi.SceneEditor.UserInterface](#)
Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Methods

HdkApi.SceneEditor.UserInterface.Menu MenuManager.CreateMenu(name)

HdkApi.SceneEditor.UserInterface.Menu MenuManager.GetMenu(name)

HdkApi.SceneEditor.UserInterface.MenuItem MenuManager.CreateMenuItem(displayText)

Example

```
#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the UserInterface classes
from HdkApi.SceneEditor.UserInterface import MenuManager, Menu, MenuItem

# Also add the System.Windows.Forms dll and import Keys so we can set a keyboard shortcut
clr.AddReference("System.Windows.Forms")
from System.Windows.Forms import Keys

class UiExample():
    def __init__(self):
        # Create a new menu
        menu = MenuManager.CreateMenu("Example Menu")

        # Create a new MenuItem
        itemDisplayText = "Example Item"
        menuItem = MenuManager.CreateMenuItem(itemDisplayText)
        menuItem.Description = "Example menu item used in the UI Example"
        menuItem.Toggleable = True
        menuItem.Shortcut = Keys.Control | Keys.Shift | Keys.M
        menuItem.HasToolBarButton = True

        # Hook up the execute event to set what happens when the menu item is pressed
        menuItem.Execute += self.exampleExecute

        # Hook up the can execute event to set whether the menu item is enabled
        menuItem.CanExecute += self.exampleCanExecute

        # Add the MenuItem to the example menu
        menu.AddMenuItem(menuItem)

    def exampleCanExecute(self):
        return True

    def exampleExecute(self, sender, args):
        print("Example executed")

example = UiExample()
```

Associated Types

- [HdkApi.SceneEditor.UserInterface.Menu](#)
- [HdkApi.SceneEditor.UserInterface.MenuItem](#)

HdkApi.SceneEditor.Utilities

Summary

Type
Description
Assembly
SynchronizationObject
Provides a synchronization object for the main Scene Editor thread, this is useful for forcing a delegate to be executed on the main Scene Editor thread, for more information about synchronization objects see: http://msdn.microsoft.com/en-us/library/system.componentmodel.isynchronizeinvoke.aspx
Home.Editors.SceneEditor.Scripting

HdkApi.SceneEditor.Utilities.SynchronizationObject

SynchronizationObject

Provides a synchronization object for the main Scene Editor thread, this is useful for forcing a delegate to be executed on the main Scene Editor thread, for more information about synchronization objects see:
<http://msdn.microsoft.com/en-us/library/system.componentmodel.isynchronizeinvoke.aspx>

Namespace: [HdkApi.SceneEditor.Utilities](#)

Assembly: *Home.Editors.SceneEditor.Scripting.dll*

Remarks

 SynchronizationObject is an advanced class which you should only have to use if you are dealing with multithreading, in general you should rarely need to use this class

Methods

System.ComponentModel.ISynchronizeInvoke SynchronizationObject.Get()

Example

```

#<copyright>
#-----
#   SCE CONFIDENTIAL
#   Home Development Kit (HDK) 1.70.0
#   Copyright (C) 2011 Sony Computer Entertainment Inc.
#   All Rights Reserved.
#-----
#</copyright>

# First import the SynchronizationObject class
from HdkApi.SceneEditor.Utilities import SynchronizationObject

# We also use the communications classes in this example, so import them as well
from HdkApi.Core.Communication import TargetManager, Target, TargetResult, TargetException

class SyncObjectExample():
    def __init__(self):
        # Create a timer
        self.timerInterval = 30
        self.timer = Timer()
        self.timer.Interval = (1000/timerInterval)

        # Initialise the TargetManager so it is ready to be used
        TargetManager.Initialize()

        # Use the SynchronizationObject to force the Elapsed callback to run on the main Scene Editor
thread
        # Normally the Elapsed callback would execute on a separate thread, but by setting the
SynchronizingObject to
        # the main Scene Editor thread timerEvent will now be called on the Scene Editor thread
instead
        self.timer.SynchronizingObject = SynchronizationObject.Get()

        self.timer.Elapsed += self.timerEvent

    def timerEvent(self, sender, arg):
        # TargetManager can only be called on the thread it was created on (the main Scene Editor
thread),
        # by setting the SynchronizingObject we know that it will be called on the correct thread
        TargetManager.ConnectToDefault()

```