Deus Ex

Software Development Kit
Editor System

07/04/00

By: Robert White, Sheldon Pacotti, Chris Norden, Al Yarusso, Scott Martin

Legal Information

Installation and use of this software constitutes agreement to the terms of this license so please carefully read its terms. If you don't agree with anything in it please delete this software from your hard drive.

This software accompanying this license remains the copyrighted property of Ion Storm and Eidos Interactive. You are entitled to use this product for your own use, but may not modify, recompile, reverse engineer, sell, rent, or lease this software to anyone else without the prior written consent of Ion Storm and Eidos Interactive.

Your license to use this software will automatically terminate immediately without receiving notice from Ion Storm and Eidos Interactive, if you fail to comply with anything in this license, you agree to destroy this software and documentation upon termination of this license.

You may use this software to create or modify levels for use with the game Deus Ex (henceforth referred to as "levels") subject to these restrictions:

Any levels you create can only work with the full, legal version of Deus Ex.

Your levels cannot contain any modifications or changes to any executable file included in this distribution or those distributed with Deus Ex.

Your levels cannot contain any illegal, trademarked, or copyrighted materials unless you obtain written consent of the trademark or copyright holders.

Any levels you create and distribute must contain your name and email address and must be distributed with a readme file that contain this sentence, "This level was not made and is not supported by Ion Storm and Eidos Interactive."

Your levels must be distributed free of charge. Neither you nor anyone else may sell, rent, lease or commercially exploit these levels in any way. You may only exchange them at no charge to licensed users of Deus Ex.

When you publicly distribute your level you automatically grant Ion Storm and/or Eidos Interactive the perpetual, royalty free right to use, modify, license and distribute your level in any way we choose.

You expressly acknowledge and agree that use of the level editor is at your sole risk. This software and documentation are provided “AS IS” and without warranty of any. Further this software is “UNSUPPORTED” by either Ion Storm or Eidos Interactive. At no time will agents or employees be required to provide the end-user assistance with the installation, configuration or operation of this software.

Under no circumstances including negligence shall Ion Storm, Eidos Interactive or their Directors, officers employees or agents can be held liable for any incidental, special, consequential or any other damages that may result from the use or inability to use this software. Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so these limitations or exclusions may not apply to you. In no event shall our total liability to you for any damages, losses, and causes of action exceed the amount paid by you for this software. If any portion of this license is unenforceable, the remaining portion shall remain in effect.

Introduction

The Deus Ex game uses the UnreadEd. The Editor classes have been modified and added to in order to support an RPG/Action game. Additional versatility in triggers, items and pawn AI are the most pronounced changes. What follows is a description of the changes we have made to the editor and how to utilize them to create a Deus Ex style mod.

Note that this is not a tutorial nor a description of the basic elements of the UnrealEd. We will not show you how to make maps or basic environments. For a tutorial, we refer you to the Unreal Website or any of the hundreds of other websites or news groups devoted to UnrealEd.

Starting up the DE UnrealEd

Once the editor has started up you will need to load the Deus Ex textures, sounds and music packages. Note that the directories textures, music and sounds need to be located within your main Deus Ex directory and preferably with their own sub-directories.

Loading the textures:

1. Click on the browse pull down window (top right side of main window) and select Textures
2. Click on load at the bottom of the same window. A dialogue box will pop up.
3. Select all of the texture packages (.utx) in the dialogue box then click open.

Loading the sounds:

1. Click on the browse pull down window (top right side of main window) and select SoundFX
2. Click on load at the bottom of the same window. A dialogue box will pop up.
3. Select all of the sound packages (.uax) in the dialogue box then click open.

Loading the music:

1. Click on the browse pull down window (top right side of main window) and select Music
2. Click on load at the bottom of the same window. A dialogue box will pop up.
3. Select all of the music packages (.umx) in the dialogue box then click open.

Implementing A Deus Ex Map

Once the map is built (Geometry created, textures and lighting added) you will want to run the map in order to see it from game perspective. You must add a *PlayerStart* somewhere on the map. It needs to be placed in the software render window so that it will be at the correct height from the ground. Secondly you will need to add a *DeusExLevelInfo* (See section on this below) icon to the map and you will need to fill in all of the appropriate fields. Once done you will need to compile the map. Press F8, then select *Rebuild Geometry*. If you have added path nodes you will need to select Lighting and then press *Paths Define.* Once this is done you can hit Ctrl-P and the map will run. Note that to se any changes in conversations or text items within the game, you must do a full recompile of Deus Ex code first.

Ladders

Unreal does not support this natively but this was added for Deus Ex, since ladders added a whole new level of accessibility to the maps. The hard code inside Deus Ex is looking for textures defined with a Group name of Ladder. Any texture package can have this group within it. To set a texture to Ladder group it has to be imported with that Group name. Once this is done, anytime the PC moves against the texture they can climb it. Currently there are two textures inside the Deus Ex texture package for Metal that are defined as Ladder group. Once this texture package is selected look under the Ladder group and you will find a plain brown texture and one that looks like a set of rungs with rails and transparent areas between the rungs. This latter one can be placed on flat polygons and scaled appropriately to look like a ladder allowing the designer to do this with a minimum amount of polys.

DeusEx Classes

New classes and alterations to existing classes were added to the Unreal system by the DE programmers. Those changes and additions are listed below. Each section contains a list of the properties and their values. Bool fields are defined as either 0 or 1 corresponding to False or True.

DeusEx Mover

The DeusExMover class is the one used most often in the game development. All the doors, glass (actually a subclass) , hatches and a few other interesting (The generator blown up in Mission 2, the tri-weld hull points on the Superfreighter, etc.) ones fall into this class. Just as in all UnrealEd movers, it will need to be created and textured first, then copy the polys or intersect the polys to the brush and select DeusEx Mover class. Then click on the mover button (left side of UnrealEd screen) and left click on the location you wish to place the mover on.

Property Name	Type	Description
bOneWay	bool	True - The door can only be opened from the direction of the arrow
blocked	bool	True - the door is currently locked
bPickable	bool	True - the door can be lockpicked
lockStrength	float	How tough the lock is to pick (0.0 = min, 1.0 = max)
bBreakable	bool	True - the door can be destroyed
doorStrength	float	How tough the door is to destroy (0.0 = min, 1.0 = max)
KeyIDNeeded	name	Key ID code which is needed to unlock door
bHighlight	bool	True - this door can be highlighted
bFrobbable	bool	True - this door can be used directly
minDamageThreshhold	int	Minimum amount of damage that must be inflicted by a weapon in one shot to seriously damage the door.
NumFragments	int	Number of fragments to spawn when destroyed
FragmentScale	float	Drawing scale of said fragments
FragmentSpread	float	Radius that the fragments are spread from the center of the door
FragmentClass	class<Fragment>	Object class of the fragment to use
FragmentTexture	texture	Texture to use on the fragments
bFragmentTranslucent	bool	True - the fragments will be translucent
bFragmentUnlit	bool	True - the fragments will be unlit
ExplodeSound1	sound	Small breaking sound
ExplodeSound2	sound	Large breaking sound
bDrawExplosion	bool	True - draw an explosion effect
bIsDoor	bool	True - this is an actual door that can be opened (instead of an elevator for example)

Deus Ex Door Defaults are as follows:
bBreakable = False
bDrawExplosion = False
bFragmentTranslucency = False
bFreagmentUnlit = False
bFrobbable = True

bHighlight = True
bIsDoor = False
bLocked = False
bOneWay = False
bPickable = True
doorStrength = 0.25
ExplodeSound1 = Sound ‘DeusExSounds.Generic.WoodBreakSmall
ExplodeSound2 = Sound ‘DeusExSounds.Generic.WoodBreakLarge
FragmentClass = Class’DeusEx.WoodFragment’
FragmentScale = 2.0
FragmentSpread = 32
FragmentTexture = none
KeyIDNeeded = none
lockStrength = 0.2
minDamageThreshold = 10
NumFragments = 16

BreakableGlass and BreakableWall

Both are subclasses of the Deus Ex mover. All the fields are the same but other default values are in them.

Glass
bBreakable = True
bDrawExplosion = False
bFragmentTranslucency = True
bFreagmentUnlit = False
bFrobbable = False
bHighlight = False
bIsDoor =False
bLocked = False
bOneWay = False
bPickable = False
doorStrength = 0.1
ExplodeSound1 = Sound ‘DeusExSounds.Generic.GlassBreakSmall
ExplodeSound2 = Sound ‘DeusExSounds.Generic.GlassBreakLarge
FragmentClass = Class’DeusEx.GlassFragment’
FragmentScale = 2.0
FragmentSpread = 32
FragmentTexture = none
KeyIDNeeded = none
lockStrength = 0.2
minDamageThreshold = 3
NumFragments = 10

Breakable Wall
bBreakable = True
bDrawExplosion = False
bFragmentTranslucency = False
bFreagmentUnlit = False
bFrobbable = False
bHighlight = False
bIsDoor =False
bLocked = False
bOneWay = False
bPickable = False
doorStrength = 0.4

ExplodeSound1 = Sound ‘DeusExSounds.Generic.SmallExplosion1
ExplodeSound2 = Sound ‘DeusExSounds.Generic.LargeExplosion1
FragmentClass = Class’DeusEx.Rockchip’
FragmentScale = 3.0
FragmentSpread = 32
FragmentTexture = none
KeyIDNeeded = none
lockStrength = 0.2
minDamageThreshold = 20
NumFragments = 16

Elevator Mover

This mover class will allow for a mover to transition between a series of key frames in a certain order. This allows for stopping at key frames and moving to the next in sequence. Note: This must be triggered by a SequenceTrigger (see below).

Field Name	Type	Description
bFollowKeyFrames	Bool	True – follow key frame sequence to destination. False – go directly to destination point by shortest route.

Multi Mover

The multi mover is used when you want the mover to go to a particular key frame for a certain time and then change to another. Up to four different times and key frames can be driven for each mover. This is useful for such things as a coplanar window with a texture denoting a light is on in the room and on the backside of the polygon is the same texture but with the lights off. The poly can be made to flip along the z axis to either side based on a situation. the light looks like it turns on and off. Note that this will requires sequence trigger to initiate this mover.

Field Name	Type	Description
bReverseKeyFrames	Bool	True – after reaching last key frame the sequence is played in reverse order. False – After reaching the last key frame the sequence starts back over with frame 1.
SeqKey1	int[4] (0-7)	Which key frame to move to for sequence 1
SeqKey2	int[4] (0-7)	Which key frame to move to for sequence 2
SeqKey3	int[4] (0-7)	Which key frame to move to for sequence 3
SeqKey4	int[4] (0-7)	Which key frame to move to for sequence 4
SeqTime1	float [4]	Move time for each key frame for sequence 1
SeqTime2	float [4]	Move time for each key frame for sequence 2
SeqTime3	float [4]	Move time for each key frame for sequence 3
SeqTime4	float [4]	Move time for each key frame for sequence 4

Decorations

Each of the following items are physical objects placed in to the world. Some have characteristics unique to them that will effect game play while some are static. This section will only deal with those whose parameters can be adjusted for game play. Once an item is placed into the world you can access its properties by right clicking on it. This brings up a menu and the properties file is the top one in the list. Select this an a large dialogue box will appear displaying all of the adjustable fields for the item. We advise you that changing some of the fields other than those listed below, could cause game or editor crashes. Be sure of the effect of the field change before you implement it. The best advice is that if you do not know what it does, don't change it.

AutoTurret and AutoTurretSmall

These are the automatic security weapon systems. They come in two sizes, but either of them can only be placed on either the ceiling or floor. The art will not support having them mounted horizontally. This is the placeable item, **not** the AutoTurretGun found under *HackableDevices.*

Field Name	Type	Description
AmmoAmount	int	Amount of ammo held in magazine of gun
bActive	bool	True - The gun initially active (default = true)
bTrackPawnsOnly	bool	True - The gun track all pawns (default =false)
bTrackPlayerOnly	bool	True - The gun only track the player (default =true)
fireRate	float	Number of seconds between each shot (default =0.25)
gunAccuracy	float	Accuracy of gun (0=very accurate, 1=poor accuracy) (default =0.5)
gunDamage	int	Damage delivered to target with each successful hit
maxRange	int	Distance in editor units (divide number by 16 = feet) (default = 512 = 32 feet)

Ceiling Fan

This is a two part object. One part comprises only the blades and axel. The second piece is the *CeilingFanMotor* which is static.

There are a couple of different skins for this accessed through the *CeilingFan* property *SkinColor* field. More important though is the field under Movement Property

Field Name	Type	Description
RotationRate	rotator	Rotation rate of fan blades. (modify only the Yaw rate)

Containers

There are several different types of these and except for the barrels they follow a common layout.

PropertyName	Field Name	Type	Description
Containers	numthings	int	Number of items of type spawned when broken
Decoration	Contents	class <Inventory>	Object spawned when container broken. NOTE: This must be an item of class Inventory.
Decoration	Content2	class <Inventory>	33% chance this object spawned when container broken. NOTE: This must be an item of class Inventory.
Decoration	Content3	class <Inventory>	33% chance this object spawned when container broken. NOTE: This must be an item of class Inventory.

Barrels have an additional important field

PropertyName	Field Name	Type	Description
Barrel1	bPreDamage	bool	True – Barrel is in a easily destroyed state and may be producing smoke or other particle effects and possible damaging nearby characters
Barrel1	SkinColor	pull down slection	SC_Biohazard – will damage player or NPC's if destroyed SC_Blue SC_Brown SC_Rusty (default) SC_Explosive – causes the most damage of all barrels when destroyed. SC_FlammableLiquid - will damage player or NPC's if destroyed

			SC_FlammableSolid - will damage player or NPC's if destroyed SC_Poison - will damage player or NPC's if destroyed SC_Radioactive - will damage player or NPC's if destroyed SC_Wood SC_Yellow

Electronic Devices

This is comprised many different types of items.

ATM

These are the cash machines of the game. They can be hacked via computer skill automatically based on script code. Note that two or more ATM's on the same map will automatically share accounts. So if funds from one for a particular account is depleted, then that same account number will be depleted on all the ATM's

Field Name	Type	Description
LockoutDelay	float	Number of real world seconds machine is locked out if hack is unsuccessful.
User List	varies	There are three fields in each of the user fields (0-7) Account Number (int) – number of account balance (int) – amount of money in the account PIN (int) – access number for account

Computers

There are three types of computers in Deus Ex and each has a specific function and capability. All of the computer types have a common field (Computers) in their properties.

Field Name	Type	Description
ComputerNode	menu	A menu list of the available nodes used in DE. This field is used by the 2D display system inside the text display code.
lockoutDelay	float	Time is seconds for computer lockout once hacking time limit is exceeded. During this interval the computer cannot be accessed and the alarm is sounding.
nodeName	string	A node ID string. This can be anything you wish
special options	varies	See table below
titleString	string	Title displayed in 2D computer access window.
titleTexture	texture	Selected texture file to be used in 2D window.
userList	varies	See Table Below

Field Name	Sub Field	Field	Type	Description
special options	0	bTriggerOnce	bool	True = event will be triggered only once.
special options	0	Text	string	Text displayed on Special options screen of computer once accessed
special options	0	TriggerEvent	string	Event name of trigger to be fired when special option is activated. NOTE: Any of the trigger types can be executed from this field. (see section on Triggers)
special options	0	TriggerText	string	Text displayed once special option is activated.

special options	0	UnTriggerEvent	string	Untriggered event name. This would apply to such items like laser triggers, particle generators, electricity generators and the like. Anything that must be turned off and does not work as a toggled event.
special options	0	userName	String	ID name of the user in the user list that can access this special option. NOTE: If no name is entered, than anyone accessing the computer can activate the special option. Also note that this must be left blank if you wish the user to be able to access this option using the computer hacking skill.

NOTE: Up to 4 possible special events can be defined for each computer. Each special event has a different sub-property number (0-3).

Field Name	**Sub Field**	**Field**	**Type**	**Description**
userList	0	accessLevel	menu	Access a pull down list of access levels. At the lowest levels the user can only access e-mail at higher levels this user could control higher level functions
userList	0	Password	string	Users password
userList	0	userName	string	Users login name

NOTE: up to 8 possible users can be listed per computer. Each has a different sub-property number (0-7).

E-mails can be read from any computer and from any level of access including hacking the system. The e-mails are read from a menu file associated with the user login name. It can be any computer on the mission that uses that login name. The syntax for the file names that we used is :

*MissionNumber*_EmailMenu_*LoginName*.txt   (02_EmailMenu_jallred.txt)

```
<EMAIL=02_Email10,Weekly Report (saved),JAllred,TMore,>
<EMAIL=02_Email11,RE: Apologies,APriest,JAllred,>
```

Email Menu file syntax

The string corresponds to: <EMAIL=*File to read*, *Subject line*, *From*, *To*, *CC* >

The actual e-mail text files look like this.

```
<DC=255,255,255>
<P><B>From:</B> JAllred//MedNet.1162.3908
<P><B>To:</B> TMore//MedNet.6629.117
<P><B>Subject:</B> Weekly Report (saved)
<P>
<P>Thomas,
<P>
<P>Another week, another report.  The triage situation here is getting tight -- so many people with the Gray Death, and we simply can't help them.  At best, we give them fluids and attempt to keep them hydrated until they crash-out, but that's just taking up space we could use for someone treatable.  A lot of them have nowhere to go, no... social support, and they're creating problems in the clinic -- security is virtually non-existent and a few times I've had to tranq someone and drag them back out onto the street.
<P>
<P>Secondary diseases on the rise: lots of tetanus, gangrene, resistant infections, advanced flu, that sort of thing. I've attached a detailed spreadsheet with a break down and supply req.  That's another thing: we missed last week's supply shipment, and we're getting low on
<P>
<P>Christ, I think they're shooting outside. I've got to go. Do what you can and let me know what to expect.
<P>
<P>Allred
```

Email file syntax

The syntax can be described as being HTML like. The first line defines the color of the printed text in RGB format. The next line contains a <P> for a hard return. A <B> to indicate start of Bold and </B> to end the bold text. Similarly <I>,</I> can be used for italics and <U>,</U> can be used for underline. Note that the block of text does not use any hard returns until a line break is needed. The text interface for DE will automatically word wrap at the appropriate point.

ComputerPublic

These are news servers for the general public. Text documents are read by this computer and displayed as e-mail style entries. Two types of files are necessary to support this computer. The first is the menu file which must be named according to the convention: *ID Number*_BulletinMenu.txt. For example the text for the public terminals in Mission 2 are 02_BulletinMenu.txt. By numbering according to mission it keeps it easy to identify each file. The menu file contains the names of the associated text files and how the header is to be displayed on the terminal screen.

```
<COMMENT>Street Bulletins</COMMENT>
<FILE=02_Bulletin01,Water Advisory>
<FILE=02_Bulletin02,Brooklyn Naval Yards Completes Sub Pen>
<FILE=02_Bulletin03,Battery Park Closed Until Further Notice>
<FILE=02_Bulletin04,Be Safe: Be Suspicious>
```

Text file for Mission 02 Bulletin Menu.

The text needs to follow the syntax as listed in the box above. The first line is a comment to simply ID the contents of the box in case of later problems. The next lines follow the syntax

```
<FILE = MissionNumber_Filename, Title to be displayed >
```

Each text file to be read uses text files named using the convention indicated in the bulletin menu file. The

```
<DC=255,255,255>
<P><B>Water Advisory</B>
<P>
<P>Residents of Manhattan and Brooklyn should be aware that last week's chlorine spill in the Hudson River now appears to have washed ashore and is interfering with the operation of at least one water processing plant.  While the impact upon the city's utilities is currently unclear, water rationing rules may need to be instituted within the next week.
<P>
<P>Under NO circumstances should any resident drink, bathe, or swim in contaminated water.  If you or someone you know comes in contact with contaminated water then go to the nearest available hospital or clinic.
```

Example of text syntax for public bulletins.

The public computer also has a field within its parameters that you must access to add the associated menu file connection.

Property Name	Field Name	Type	Description
ComputerPublic	bulletinTag	string	Name of bulletin file

ComputerPersonal

this is the desktop system of the DE world. Usually used only as a means of e-mail reception it has similar properties of all of the computers and can activate special options triggering outside events (i.e. Shutting off the coolant to the generator in the NYC warehouse and blowing it up, Signaling the NSF from the old NSF headquarters building, etc.)

ComputerSecurity

The security computer or security panel has all the features of the other computers plus the ability to access the camera systems, locked doors, and auto turrets. These features are accessible through the Computer Security field , View property.

Field Name	Type	Description
cameraTag	text	Camera tag name
doorTag	text	Door tag name
titleString	text	Text for camera view seen in 2D window
turretTag	text	Auto turret tag name

NOTE: Up to three views can be used per security computer (0-2). Also any door or auto turret designed to be used by the security computer must have an associated camera tied with it (preferably with a view of the turret or door).

Hackable Devices

Any device that requires using the multitool and associated technical skill is considered a hackable device (computer hacking falls under the computer skill and does not require the multitool). All of these devices share a common field (HackableDevices) described below.

Field Name	Type	Description
bHackable	bool	True – this device can be hacked, False – hack strength set to infinite.
hackStrength	float	Strength value of device defined as a percentage (0.1 = 10%, 1.0 = 100%)
UntriggerEvent	string	There are four of these sub-properties. Up to four unTrigger events per device. The field will contain the unTrigger tag name for the item you wish to unTrigger.

AcousticSensor
This is set off by any explosion or gun shot within range. Range is determined according to parameters within the Event Manager code of the AI and is based on both volume and range from sensor.

AlarmUnit
This is used as an alarm trigger by NPC's. An NPC who is set to activate alarm before or prior to fleeing a battle will seek out the nearest alarm panel or one that has been tagged and they are set to go to the tagged alarm (see NPC Orders property) and activate it. This sounds an alarm and will trigger other NPC's to an alert state if they are set to react to the alarm and are within hearing range of it.

Field Name	Type	Description
alarmTimeout	int	The number of seconds the alarm will sound before shutting off automatically (default = 30)
Alliance	string	This is the named tag of NPC alliance that will be alerted by the alarm. If set to default (none), then all NPCs in audio range set to react will react. If an alliance is set, then only those NPC of that alliance will react. Useful if in areas where a lot of neutral NPCs wouldn't react to the alarm but guards would.

ControlPanel
Used as a hacked off switch for a lot of things in the game, especially laser triggers and electric fields.

Keypads
There are three different versions of the keypad, but all function identically, only the art is different. Using any keypad activates a 2D interface screen.

Field Name	Type	Description
bToggleLock	bool	True – the door will only unlock. False – the door will activate and open.
FailEvent	string	The tag of a failure event. Often used with a triggered Ambient sound to set off an alarm.
failureSound	sound	Defaults to Generic.Buzz1
successSound	sound	Defaults to Generic.Beep2
validCode	string	This is a string, but limited to those as seen on a telephone pad (numbers 0-9, #, *)

RetinalScanner
A device that works like a keypad. Unfortunately it is set to accept anyone using the device. The only field is a text string indicating that clearance is granted.

SecurityCamera
Used in conjunction with the security camera to provide views of game areas located on the same map with the security computer. Note that these devices do not work across different maps. The camera is considered a hardened target and takes a lot of damage delivered at once to take it out (rifle at high mastery, an explosive, etc.)

Field Name	Type	Description
bActive	bool	True – camera is running when first encountered.
bNoAlarm	bool	True – If the camera is activated by player presence in FOV, then an alarm will sound.
bSwing	bool	True – Camera will pan within swing angle parameters
cameraFOV	int	The field of view angle of the camera. This is a binary angle measurement (65536 = 360 degrees)
cameraRange	int	Distance measure in game units (divide by 16 for feet)
swingAngle	int	The angle the camera rotates through for the swing. The camera will swing this angle both + and – of the point it positioned at in the editor. . This is a binary angle measurement (65536 = 360 degrees)
swingPeriod	float	The number of seconds that the camera takes to complete a full swing cycle.

Phone

The device is purely a decorative item in Deus Ex but it has some variable parameters.

Field Name	Type	Description
AnsweringSound	menu	A selection of different audio files are played if item is used
ringFreq	float	A randomizing variable giving basic chance that the player will here the phone ring when passing within range of the phone. (default = 0.01 = 1%)
RingSound	menu	A selection of two different ring audio files.

TAD

This is an answering machine. It has a couple of parameters to make it beep, drawing attention to itself. To make it work though a conversation file will need to be created for it. The file name (BindName) must then be placed in the *Conversation* property, *BindName* field. Then whenever the TAD is used the conversation file is played.

Field Name	Type	Description
beepInterval	float	Number of seconds between playing the beep sound
beepSound	sound	Defaults to Generic.Beep5

VendingMachine

Script code generates either a drink or a soy food snack based on which skin is selected from the *VendingMachine* field in the item properties.

Field Name	Type	Description
SkinColor	menu	Choice of either Drink or Snack. Changes both image and functionality.

Faucet

Standard sink style faucet seen all through the game.

Field Name	Type	Description
bOpen	bool	True – Is the faucet in an open position and running

FlagPole

Field Name	Type	Description
SkinColor	menu	A selection of possible skins for the flag pole. The include China, France, US Presidential, UNATCO, USA

HangingDecorations

There are several items in this category. All will react to being struck by swaying back and forth around the pivot (which is at the top of their connecting cord). Some, like the shop light, are connected by two points so they only sway in one direction.

Field Name	Sub Field	Type	Description
bFixedSwaying	-	bool	True – Item will be swaying based on FixedSwayDirection Field parameters when first encountered
bNoPitch	-	bool	True – Restricts motion of hanging item around this direction if item is bumped or damaged.
bNoRoll	-	bool	True – Restricts motion of hanging item around this direction if item is bumped or damaged.
bRandomize	-	bool	True – randomizes rotation axis, period and sway amplitude.
FixedSwayDirection	Pitch	int	Amplitude of pitch direction. This is a binary angle measurement (65536 = 360 degrees)
FixedSwayDirection	Yaw	int	Amplitude of yaw direction. This is a binary angle measurement (65536 = 360 degrees)

FixedSwayDirection	Roll	int	Amplitude of roll direction. This is a binary angle measurement (65536 = 360 degrees)
FixedSwayPeriod	-	float	Time in seconds for one full cycle of sway to occur.

InformationDevices

There are several in this class, but the properties for each are the same. The information devices are: *BookClosed, BookOpen, DataCube,Newspaper,* and *NewspaperOpen*.

Field Name	Type	Description
imagleClass	class<Inventory.DataVaultImage>	A specially prepared image for 2D display (i.e. Satellite image of the warehouse area)
textTag	string	Name of the information text file.

The text files for the information devices follow the same syntax as that of the e-mails described earlier (see Computers). The naming convention for the files in Deus Ex is as follows:

Book (open or closed): *MissionNumber*_Book*IDNumber*.txt (02_Book01.txt)
DataCube: *MissionNumber*_Datacube*IDNumber* (02_Datacube01.txt)
Newspapers (open or closed): *MissionNumber*_Newspaper*IDNumber* (02_Newspaper01.txt)

OutdoorThings

There are several items in this category. A few of them have a property associated with their name that allows for changing the skin on the object. Those with this property are: *Bushes3*, *CarBurned*, *CarStripped*, *FirePlug*, *Tree1*, and *Tree2*

In addition, Tree1, Tree2, Tree3 and Tree4 have another property called Tree

Field Name	Type	Description
soundFreq	float	a random chance of the tree to produce a wind noise when encountered (default = 0.2 = 20%

Poolball

There is a single property called Poolball that allows for changing the pool ball skin. The skins are SC-1 through SC_15 and SC_Cue.

ShowerFaucet

This object must be paired with an associated ShowerHead. The tag name of the ShowerFaucet and the ShowerHead must be the same in order for them to work.

Field Name	Type	Description
bOpen	bool	True – Shower is running when the player firsts encounters it.

Effects

ElectricityEmitter

This is a sub class of LaserEmitter (By the way, if you try and place a laser emitter, nothing will happen). The electricity emitter produces electrical bolts that will collide with the character and can cause damage. This is a directional device. Once placed an arrow is visible in the overhead and side views of the editor (but not the software render view). The arrow will need to be rotated to point in the direction that you wish the electrical bolts to go. This item can be triggered on, but must be unTriggered to turn it off (with something like the ControlPanel).

Field Name	Type	Description
beamTexture	texture	Texture. Defaults to Effects.Electricity.Nano_SFX
bEmitLight	bool	True – Emitter will flash with blue light every time electricity flickers on.
bFlicker	bool	True - Emitter flickers on and off . Flickering is randomized with max limit as defined by flickerTime
bInitiallyOn	bool	True - Emitter is running when player encounters it
DamageAmount	int	Damage applied to character
damageTime	float	Time in seconds that damage will occur to character remaining in contact with electric field. Default = 0.2
flickerTime	float	Maximum time in seconds between on and off cycle. Default = 0.02
randomAngle	float	Deflection angle from center point (defined by the Arrow). This is a binary angle measurement (65536 = 360 degrees)

ParticleGenerator

This is very similar to the one already being used in the standard Unreal engine, but there are few additional capabilities. One of the main things is that it can be turned on and off via triggers and unTriggers.

Field Name	Type	Description
attachTag	string	Tag name of the mover to attach the generator to
bAmbientSound	bool	True – generator will create a sound when particles are spawned
bFade	bool	True – particles will fade as the move farther away from the generator
bGravity	bool	True – particles will fall downward after being spawned
bIntiallyOn	bool	True – generator is running when player first encounters it.
bModulated	bool	
bParticlesUnlit	bool	True – particles will appear bright as if self illuminated.
bRandomEject	bool	True – particles will be ejected at random angles (<180) from generator
bScale	bool	True – particles will scale larger as they move away from generator
bTranslucent	bool	True – particle texture will be rendered as translucent
bTriggered	bool	True – generator can be turned on and off from a trigger
checkTime	float	Time in seconds that the system checks for the possibility of generating a new particle. Default = 0.1
ejectSpeed	float	The speed at which particles are emitted. Default = 10.0
Frequency	float	The base chance that a particle will be generated during the checkTime. Default = 1.0
numPerSpawn	int	The number of particles spawned during a successful generation at checkTime. Default = 1
particleDrawScale	float	The size of the particles in relative terms (2=2x norm, etc.) Default = 0.10
particleLifeSpan	float	The duration that a particle will remain on screen. Default = 4.0
particleTexture	texture	The texture primitive used for the particle. Default = none (actually uses smoke texture)
RiseRate	float	The speed at which the particles will rise from the generator. Note: If gravity is turned on (bGravity=true) then this will not work. Default = 10.0
SpawnSound	sound	The sound made by the particle ejection. bAmbientSound must be true for this to work. Default = none
spewTime	float	Once a successful checkTime is achieved and the Frequency randomization indicates that a particle will generate, this is the time in seconds that a particle will be emitted. Default = 0 (instantaneous)

WaterDrips

All the parameters are the same, but the defaults are different. Speed of drips can be adjusted by increasing or decreasing frequency and checkTime, but other parameters should be left alone. When positioning this generator. Adjust the arrow to point in the downward direction and place the generator directly on a ceiling or the underside of any other surface.

PawnGenerator

These generators will create NPC creatures. There is only basic default behavior AI possible. The creatures or people are generated only when the player does not have line of sight on the generator. The unit will maintain a constant balance of a certain number of these NPC's. If a generator is set to create 5 pigeons and the player kills 2 and runs away, then the generator will create 2 more to maintain the balance in the area.

All of these generators have the same properties, but with different defaults. The types of generators are *Fish2Generator. FishGenerator, FlyGenerator, PigeonGenerator, RatGenerator,* and *SeagullGenerator.* Note that there is a PedestrianGenerator listed in the class files, but it is non-functional.

Field Name	Type	Description
ActiveArea	float	The area in which the player must be within for the unit to actually spawn creatures
Alliance	string	The alliance linking all of the creatures together (see the section on AI)
bGeneratorIsHome	bool	True - The physical generator is used as the home base point for determining the range at which the creatures can wander to.
bTriggered	bool	True – The generator can be turned on and off from a trigger.
MaxCount	int	The maximum count of creatures the generator will maintain
Orders	string	The AI command inherent to all creatures spawned by this generator. Default = Wandering. See AI for other possible choices.
OrderTag	string	An item tag name used by the creature in conjunction with some of the AI orders. See section on AI. Default = none.
PawnHomeExtent	float	Distance home point is located from generator if bGeneratorIsHome = false.
Radius	float	Distance that creatures can move away from home point.

ProjectileGenerator

This is a special version of the *particleGenerator* that creates a damaging object. There are three additional fields found in the projectile generator not found in the *particleGenerator*

Field Name	Type	Description
bSpewUnseen	bool	True – The projectile generator will be spewing even though the unit may be hidden from view.
ProjectileClass	class <Projectile>	Particle primitive spawned for the effect.
ProjectileLifeSpan	float	Time in seconds that projectile exists once spawned
SpawnSoundRadius	float	Distance from generator that spawn sound can be heard
WaitTime	float	Time between spawning of projectiles

The damage from the projectile is set by parameters in the class definition for the projectile type, not by the generator itself.

TrashGenerator

This is another of the unique particle generators. This one produces items of trash that appear to blow along in the wind. These tend to be a bit 'hoggish' on the frame rate so use them sparingly. When placing the generator use the directional arrow as described above. The trash will move in this direction. If you place it at an angle other than horizontal with the ground it may look funny when the item of trash blows up into the air.

Field Name	Type	Description
Frequency	float	Chance each 10th of a second that a trash item will be generated.
TrashType	menu	Two possible items selectable. Paper or tumbleweeds
WindSpeed	float	The speed at which the trash is carried away from the generator.

Info

This section of classes deals with game information tools. One of the larger sections is the mission Script section which has categorized sub-classes for each of the Deus Ex missions. All of these script files are done with Unreal Script.

DeusExLevelInfo

These items must be included with every map generated. It contains the necessary information for the mission scripting to tie into the map. It will also contains the true north compass indicator and start up message string fields.

Field Name	Type	Description
bMultiPlayerMap	Bool	True – This map is set for multiplayer and will access future support scripts for multiplayer
MapAuthor	String	map authors name
MapName	String	file name of the map
Mission Location	String	geographic location of the map
MissionNumber	Int	This is the number of the mission. This is limited to the numbers between 16 and 97. The numbers 98 and 99 are reserved for the end games and pre game sequences. 0-15 are the actual Deus Ex game missions.
Script	class <Info.MissionScript>	This links the associated mission scripting with the map
startupMessage	string [3]	There are four sub-fields here. This is the information that is typed onto screen as the map loads. Note, however that it will run every time this map is accessed, not just the first time.
TrueNorth	Int	This value will determine the direction of true north for the in game compass. The map designer needs to determine this based on the overhead view of the map. If north is the top of the overhead map, then north = 49152 If north is the right of the overhead map, then north = 0 If north is the bottom of the overhead map, then north = 16384 If north is the left of the overhead map, then north = 32768 If it's not a simple 90 degree offset, you can calculate it by taking the number of degrees clockwise from 0 (which points to the right) and multiplying it by 182.

ZoneInfo

This unti is used for defining differences in zone areas. A few tweaks were added form the original Unreal zones. Important properties for DE are listed below.

Property Name	Field Name	Type	Description
ZoneInfo	bFogZone	bool	True – this zone will render volumetric lights and fog.
ZoneInfo	bGravityZone	bool	True – This zone uses the ZoneGravity parameters. False – there is no gravity in this zone.
ZoneInfo	bKillZone	bool	Tue – Zone will kill any players entering it.
ZoneInfo	bNeutralZone	bool	True – Player cannot take damage while in this zone
ZoneInfo	bPainZone	bool	True – This zone damages the player. **NOTE:** IF the DamagePerSec field is non-zero then the zone info can be triggered on and off to cause or stop pain.
ZoneInfo	bWaterZone	bool	True – This zone is under water

ZoneInfo	DamagePerSec	int	Damage accrued by player every second
ZoneInfo	DamageString	string	Text displayed when damage occurs
ZoneInfo	DamageType	string	Type of damage. Will access icon to be displayed on HUD indicating damage. Valid types are: Shot, Exploded, TearGas, PoisonGas, HalonGas, Radiation, Flamed (catches player on fire), Burned, Shocked, NanoVirus, EMP, Drowned, Stunned.
ZoneInfo	ZoneFluidFriction	float	Fluid friction coefficient (normal = water =1.2)
ZoneInfo	ZoneGravity	vector	Gravity within zone
ZoneInfo	ZoneGroundFriction	float	Ground friction coefficient (normal = 4.0)
ZoneInfo	ZonePlayerEvent	string	Event triggered each time player enters zone
ZoneInfo	ZoneTerminalVelocity	float	Maximum speed any actor in zone can achieve
ZoneInfo	ZoneVelocity	vector	Velocity vector of zone. Useful for strong water or air currents.
Zone Light	AmbientBrightness	byte	Zone brightness
Zone Light	AmbientHue	byte	Zone hue
Zone Light	AmbientSaturation	byte	Zone saturation
Zone Light	EnvironmentMap	Class<Texture>	Default texture for zone
Zone Light	TextUPanSpeed	float	Horizontal panning speed of textures
Zone Light	TexVPanSpeed	float	Vertical panning speed of textures
Zone Light	ViewFog	vector	Color vector for fog. Only works if bFogZone=true.
Reverb	bRayTraceReverb	bool	True – raytrace the reverb effects within this zone
Reverb	bReverbZone	bool	True – Within this zone reverb is applied to all sounds
Reverb	CutoffHz	int	Reverb effects are cut off of at this frequency
Reverb	Delay	int [6]	Time in milliseconds that each reverb stage is delayed
Reverb	Gain	int [6]	Volume for each of the reverb stages
Reverb	Master Gain	int	Gain for all reverb effect
Reverb	SpeedOfSound	float	Speed of sound within zone.

Inventory

All items of this calls are capable to be picked up by the player and placed into inventory. No other class allows this. Any items mod developers wish to have within the game for players to pick up should be placed into this class. All items have the Pickup property in common.

Field Name	Type	Description
bAutoActivate	bool	Obsolete
bCanActivate	bool	Obsolete
bCanHaveMultipleCopies	bool	True - item can be stacked in inventory
ExpireMessage	String	Text displayed once item is used up if it is charged.

DataVaultImage

These consist of all of the images available that can be inserted into the players DataVault. Some of these include the LibertyIslandMap, AmbrosiaFlyer, etc.

Ammo

Each ammo type has a set of common fields. Some of these were intended for future use but were never implemented.

Field Name	Type	Description
AmmoAmount	int	Total number of rounds in this item.
MaxAmmo	int	Number of rounds of this type that the player can possibly carry
ParentAmmo	-	Not used
UsedInWeaponSlot	-	Not used

DeusExPickup

These are all items not associated with weapons or ammo.

AugmentationCannister

Field Name	Type	Description
AugmentationCannister	class <Augmentation>	The name of the augmentation class type is typed in here. There is no menu set so user must locate correct aug type syntax from class. NOTE: There is a slot for two aug types to be listed. The augmentation class supports the decision process and requires the med bot to activate.

ChargedPickup

These items have a limited life span based on time. The time factors are decided in the Unreal script for the item. Time factors and effectiveness are varied based on if there is a skill associated with the particular item as defined by one of the charged pick up fields.

Field Name	Type	Description
ActivateSound	sound file	Sound when item is first used
bOneUseOnly	bool	True – the item can be used more than once.
DeActivateSound	sound file	Sound when item is used up
LoopSound	sound file	Sound while item is running
skillNeeded	class <Skill>	Skill that will modify the effectiveness of the item.

Liquor40Oz

Field Name	Type	Description
SkinColor	menu	Selection of skins for the bottle model

NanoKey

Field Name	Type	Description
KeyID	string	Name of the key. This must match the string for KeyID for any door this is to open. See DeusEx mover.
SkinColor	menu	Selection of skins for the nano key model.

WeaponMod

There are no fields that actually need to be changed for these. All variables for these items are coded within UnrealScript.

Weapon / DeusExWeapon

All Weapons in DeusEx have this property. Within are many fields to adjust weapons to fit just about any criteria needed for development of weapons in the DE universe.

Field Name	Type	Description
AccurateRange	int	Range at which the accuracy of the weapon suffers no penalties
AmmoNames	class <Ammo> [3]	3 sub fields containing classes of ammo allowable with this specific weapon.

AreaOfEffect	menu	Area of effect type. Current selections are point, sphere and cone.
baseAccuracy	float	Base accuracy of this weapon type
bAutomatic	bool	True – weapon continues to fire as long as weapon fire button is held down.
bCanTrack	bool	True – weapon has target tracking capability
bHandToHand	bool	True – weapon is defined to be a melee’ weapon
bHasLaser	bool	True – weapon has laser accuracy mod
bHasMuzzleFlash	bool	True – weapon displays a muzzle flash animation
bHasScope	bool	True – weapon has scope
bHasSilencer	bool	True – normal weapon firing noise is eliminated.
bPenetrating	bool	True – weapon has body penetrating damage type.
Concealability	menu	Can weapon be noticed by AI or other means. Selection of concealment is: None, Visual, Metal, All
EnemyEffective	menu	Weapon effectiveness versus different enemy types. Selection is: All, Organic, Robot
EnviroEffective	menu	Environments weapon is effective within. Selection is: All, Air, Water, Vacuum, AirWater, AirVacuum, WaterVacuum
GoverningSkill	class <Skill>	Skill class used to determine effective ness of player using this weapon
HitDamage	int	Damage done directly by weapon (note ammo carries the damage parameters)
LockedSound	sound	Sound weapon emits once locked on target.
LockTime	float	Time in seconds for target to be tracked before lock occurs.
LowAmmoWaterMark	int	Point at which ammo indication on HUD indicates that weapon resource is low.
maxRange	int	Maximum range of weapon. Beyond this point it cannot hit. Between this range and the AccurateRange field, damage is reduced in effectiveness.
NoiseLevel	float	How loud the weapon is. This is used by the AI to determine if opponents can hear the weapon firing.
ProjectileName	class <Projectile> [3]	
recoilStrength	float	Recoil value (none = 0, max = 1)
reloadTime	float	Time in seconds to completely reload the weapon
ScopeFOV	int	Angular field of view from scope (Default =10)
ShotTime	float	Time in seconds between successive shots
StunDuration	float	Time target is stunned by weapon
TrackingSound	sound	Sound used by weapon when target is tracked.

Keypoint

AmbientSoundTriggered
These are used so that sound effects can be started and stopped from triggers. Tag name of AmbientSoundTriggered must be the Event name of the trigger.

Field Name	Type	Description
bActive	bool	True – Sound is running when first encountered by player
bTriggerOnlyOnce	bool	True – Sound is triggered only once.

PickUpDistributor
This is used to distribute nanokeys to NPC inventories. It initiates when the map first starts up. It will only distribute to NPCs on the same map as the distributor.

Field Name	Sub-Field	Type	Description
NanoKeyData	Description	String	Description of the keys purpose (i.e. Maggie's back door)
	KeyID	String	Name of Key. Used by mover in DeusExMover property. See DeusExMover section.
NanoKeyData	ScriptedPawnTag	String	Tag name of NPC to be distributed to
NanoKeyData	SkinColor	Menu	Selection of key appearances

RandomEvents
Once triggered this will initiate a series of random events based on chance defined by the frequency field.

Field Name	Type	Description
bLOSRequired	bool	True – Events are only triggered if player can see this location
Events	string [8]	Tag names for 8 possible event triggers
Frequency	float [8]	Chance for each event to occur. Percentage define by 1.00 = 100% (Default = 0.125=12.5%). Chance for success is checked every second. Each frequency corresponds to the same numbered event above.

RandomSounds
This allows for a group of sounds to be played at random intervals in order to bring life to an area. Up to eight different sound effects can be run by the RandomSound generator.

Field Name	Type	Description
bAmbient	bool [8]	True – This sound has been created as a continuously looped sound effect.
bFade	bool [8]	True – This sound will be faded in and then faded out. Only works for sounds where bAmbient = true.
bFakeDoppler	bool [8]	True – A Doppler effect will be added to the start and end of the sound effect. . Only works for sounds where bAmbient = true.
Frequency	float [8]	Percentage chance that the sound will be played. Check is made every second. (Default = 0.125 =12.5%)
maxDuration	float [8]	Maximum time, in seconds, that the sound will be played. Only works for sounds where bAmbient = true.
minDuration	float [8]	Minimum time, in seconds, the sound will be played. Only works for sounds where bAmbient = true.
Pitch	int [8]	Adjustment to the pitch of the sound. (Default = 64)
Sounds	string <Sound File> [8]	Name of the sound file to be played
Volume	int [8]	Volume of the sound to be played.

Two other factors will be useful for this sound generator. Under the sound property, insert any sound into the AmbientSound field. This will allow the radius of effect to be visible in the editor. It doesn't matter what is placed here since it will not be played. Also the SoundRadius field under Sound property can be changed. This is the radius within which the random sounds will be heard.

SequenceEvents
This is a controller that will fire a series of events off. Each will be fired with a specified delay between each subsequent event and the events can be repeated if desired.

Field Name	Type	Description
bLoop	bool	True – Sequence will repeat
bLOSRequired	bool	True – Player must be able to see this location for the sequence to run.
Delays	float [8]	Time in seconds between each event.
Events	String [8]	Tag name of each event to trigger.

RestartTime	float	Time delay before restarting sequence. bLoop must be true for this to work.

Navigation Point

These classes are used by the NPC's and the player to negotiate the geometry of the world.

HidePoint

There are no parameters to set for this class. It is a very important navigation point for the AI of Deus Ex. Under certain conditions the NPCs will flee, seeking hide points to gather at. If these are not available the NPC will attempt to conceal themselves from the players LOS by the first available piece of geometry. Most of the time this does not look good or is not logical behavior. The best thing is to have several of these around a map in viable locations for NPCs to want to go to in case of an emergency. When a frightened NPC reaches a HidePoint, he will face the same direction that the HidePoint is facing.

MapExit

This is used as a link to each mission. Teleports were used between maps within a mission. We used the map exit as a one way directional teleport that follows the plot thread of the story. We usually triggered these off of the Black Helicopter in Deus Ex. That Helicopter would then trigger the map exit.

Field Name	Type	Description
bPlayTransition	bool	True – play cinematic transition
cameraPathTag	string	Tag name of camera path used in cinematic transition
DestMap	string	Name of map to go to.

Pathnodes and PatrolPoints

Rules For Placing Pathnodes in UnrealEd

When adding pathnodes, always click on the floor in the 3D view while pressing "A" -- this puts the pathnodes at the right height above the floor for a normal-sized NPC. (Pathnodes for short NPCs, like cleaner bots, will have to be moved lower to the ground.)

Place your pathnodes less than 700 units apart. Pathnodes on stairs should be placed less than 350 units apart.

Always place your pathnodes on a direct line of sight with other pathnodes. Unreal can't connect pathnodes unless there's a direct line of sight between them.

Be sparing in your use of pathnodes. Place the minimum number of pathnodes needed to get around.

Pathnodes should be placed in corners or at decision points. Avoid putting a pathnode directly between two other pathnodes unless the distance between the two other pathnodes is greater than 700 units.

After you've place pathnodes, hit F8, select the "Lighting" tab, and click on the "Paths Define" button. This will generate the connections between pathnodes.

You can see the connections by selecting the View->Show Paths menu option in any of the view windows in UnrealEd. Paths shown in blue are clear paths -- any NPC of normal size will be able to navigate along these paths. Paths shown in red are narrow paths. Some NPCs will be too big to move down these paths. Avoid creating red paths if possible. (Sometimes it can't be avoided.)

You can sometimes eliminate red paths by moving a pathnode to a more accessible location with a clearer line of sight, and then rebuild the paths.

PatrolPoints are points that an NPC can patrol along. NPCs will always move in a predetermined order along these PatrolPoints. Once an NPC has reached a PatrolPoint, he goes to the next PatrolPoint in the list, which you can set via the NextPatrol property (under PatrolPoint). NextPatrol represents the tag of the next patrol point.

HomeBases are points which the NPC tries to keep in sight at all times while wandering. HomeBases are ignored when the NPC is fleeing or attacking. NPCs will try not to move out of sight of the HomeBase, and they will try to stay within a certain radius of the HomeBase (which you can set via the Extent property). An NPC's HomeBase can be set via that actor's OrderTag property.

Pawn

Pawns are the NPCs in Deus Ex. The class of most use is ScriptedPawn. Beneath this are subclasses of Animal, HumanCivilian, HumanMilitary, HumanThug, and Robot.

Useful Properties for all Pawns.

Property Name	Field Name	Type	Description
AI	bAlwaysPatrol	bool	True – Even if NPC is not in player LOS, the NPC will continue moving on its patrol route
AI	bCanSit	bool	True – NPC can be made to sit down if ordered. Animations must be made if this is to work.
AI	bCower	bool	True – NPC will cower if fear reactions occur
AI	bEmitDistress	bool	True – Once the NPC is in a fleeing or cowering state they will emit distress that other AI can react to.
AI	bHokeyPokey	bool	True – If NPC is set to dance (animations must exist for the NPC) then NPC will on occasion spin.
AI	bKeepWeaponDrawn	bool	True – NPC will keep weapon in hand
AI	bLeaveAfterFleeing	bool	True – Once NPC is fleeing and gets beyond player LOS, the NPC will be deleted from map.
AI	bLikesNeutral	bool	True – NPC will respond to neutral NPCs as if they were allied together.
AI	bPlayIdle	bool	True – When NPC is standing for a period of time they will play idle animations
AI	bShowPain	bool	True – NPC shows pain flinch animation when damaged
AI	bTickVisibleOnly	bool	Obsolete
AI	bUseFirstSeatOnly	bool	True – If NPC is set for orders sitting, then it will move toward a designated or closest chair. If that chair somehow becomes unavailable, then it will not seek out another chair to sit in, but will instead revert to standing orders.
AI	EnemyTimeout	float	time in seconds that NPC will remain in a seeking state before giving up
AI	HearingThreshold	float	Acuteness of hearing. The closer the threshold is to zero, the better the hearing.
AI	MaxProvocations	int	Number of times this NPC can be damaged by friendly units before they turn hostile to that attacker.
AI	PeripherealVision	float	Acuteness of side vision
AI	RaiseAlarm	menu	Choice of raising alarm before attacking, when fleeing, or never
AI	SightRadius	float	Obsolete
AI	Skill	float	Obsolete
AI	visibilityThreshold	float	Acuteness of vision or how easily they can notice a player. The closer this number is to zero, the better the NPC's sight is.

Alliance Property

Field Name	Sub-Field Name	Type	Description
Alliance		string	Alliance name this NPC belongs to

InitialAllinaces [8]	AllianceLevel	float	Only viable choice are : -1 = Hostile toward 0 = Neutral toward 1 = Friendly toward
	AllianceName	string	Name of alliance to apply the AllianceLevel to
	bPermanent	bool	True – other game conditions will not cause this alliance to be altered (i.e. they are friendly and you shoot them many times in the back)

Property Name	Field Name	Type	Description
Combat	bAimForHead	bool	True – the AI will try for head shots. Note that these are fast kills on the player
	baseAccuracy	float	Rating on how good the NPC is at hitting a target 0 = exceptional, 1 = Poor
	bAvoidAim	bool	True – NPC will dodge to avoid getting shot.
	bCanStrafe	bool	True – NPC can slide left and right while firing
	bDefendHome	bool	True – NPC will stay near home base and defend that point.
	bMustFaceTarget	bool	True – NPC must turn towards target before firing

Property Name	Field Name	Type	Description
Conversation	BarkBindName	string	Tag name for barks. If the character already has barks specifically generated for them in ConEdit, then this can be left blank and the bark system will use the BindName. There are some generic barks for the DE system which use Male, Female, and Child for the BarkBindName
Conversation	BindName	string	This is the name of the conversation file attached to this character.
Conversation	ConStartInterval	float	Time in seconds between separate conversations being played. (Default = 5.0)
Conversation	FamiliarName	string	Name displayed on user interface when character has been met once
Conversation	UnFamiliarName	string	Name displayed on user interface if character has never been met.

The next three properties drive the AI reaction state of the NPCs. The first one, Fear, creates a distress reaction in the NPC based on the fear type. The NPC will flee or cower when a fear reaction occurs. Most hostile or combat NPCs do not have any fear parameters set as True. The seconds is their Reaction. Reaction defines the type of situation they will react to or if they will ignore it. the final one is their Stimuli. These are situations that will cause a hostile reaction in the NPC. Unless the NPC has an alliance set with bPermanent = true, this can cause a reversal of their alliance level, making them hostile to the NPC or player that activated the Stimuli.

Property Name	Field Name	Type	Description
Fear	bFearAlarm	bool	True – Alarm creates distress reaction in NPC
Fear	bFearCarcass	bool	True – NPC visually seeing a carcass creates distress reaction in NPC
Fear	bFearDistress	bool	True – Other NPCs' distress creates distress reaction in this NPC
Fear	bFearHacking	bool	True – Hacking in view creates distress reaction in NPC
Fear	bFearIndirectInjury	bool	True – Damage from indirect means (gas, explosives, etc.) creates distress reaction in NPC
Fear	bFearInjury	bool	True – Damage from blows or other means directed at the

			NPC creates distress reaction in NPC
Fear	bFearProjectiles	bool	True – Grenades or other projectile weapons in view of NPC, creates distress reaction in NPC
Fear	bFearShot	bool	True – Being shot creates distress reaction in NPC
Fear	bFearWeapon	bool	True – Seeing a weapon in the NPC or player hand creates distress reaction in NPC

Property Name	Field Name	Type	Description
Reactions	bReactAlarm	bool	True – NPC will react to an alarm going off within its auditory range
Reactions	bReactCarcass	bool	True – NPC will react to a carcass within its visual range
Reactions	bReactDistress	bool	True – NPC will react to distress from other NPCs
Reactions	bReactFutz	bool	True – NPC will react if player picks up or uses an item within their visual range. **NOTE:** The item must have its Advanced Property bOwned = true. If the item is not flagged as such the NPC will not react to it.
Reactions	bReactLoudNoise	bool	True – NPC will investigate a loud noise within its auditory range (i.e breaking a bottle, window, blowing up a LAM)
Reactions	bReactPresence	bool	True – NPC reacts to the players presence
Reactions	bReactProjectiles	bool	True – NPC reacts to projectiles thrown near it
Reactions	bReactShot	bool	True – NPC reacts to being shot.

Note for Stimuli: If an NPC goes into hate or hostile mode due to a stimuli happening they will attack unless they are weaponless or out of ammo. In that case they will flee the area.

Property Name	Field Name	Type	Description
Stimuli	bHateCarcass	bool	True – NPC will turn hostile when they see a corpse
Stimuli	bHateDistress	bool	True – NPC will turn hostile when they sense distress from another NPC.
Stimuli	bHateHacking	bool	True – NPC will turn hostile if they see the player hacking a computer or ATM.
Stimuli	bHateIndirectInjury	bool	True – NPC will turn hostile if they are damaged indirectly (gas, explosives, etc.)
Stimuli	bHateInjury	bool	True – NPC will turn hostile if damaged by any means.
Stimuli	bHateShot	bool	True – NPC will turn hostile if they are shot.
Stimuli	bHateWeapon	bool	True – NPC will turn hostile if a weapon is pulled out or displayed in their presence.

Inventory

This property sets up the items the NPC is carrying. This must include weapons and ammo if they are to fire on others. The ammo and weapon also have to match up. NOTE: The secondary ammo types currently are not useable by NPCs (i.e. Sabot rounds, 20mm, etc.)

NPCs will utilize distance weapons if they have range, but within a certain radius they will use melee' weapons or they will try to move away for distance.

Field Name	Sub-Field Name	Type	Description
InitialInventory[8]	Count	int	Amount of the item NPC has
	Inventory	class<Inventory>	Inventory class

Orders

Property Name	Field Name	Type	Description

Orders	AlarmTag	string	Alarm name tag of a specific alarm this NPC will seek out if set to raise alarm.
Orders	bFixedStart	bool	Obsolete
Orders	HomeExtent	float	The range the character will roam away from the HomeTag point (or Start). Used if orders are wandering, but this will be superceded if NPC goes into seeking or attacking mode.
Orders	HomeTag	string	Name of a radius point for NPC to reference around. If set to “start”, then this is the initial location the NPC begins at.
Orders	Orders	Selection	Order type . See AI state information below.
Orders	OrderTag	string	reference point for order. Could be a patrol point tag name, that of an NPC, chair, etc.
Orders	SharedAlarmTag	string	Obsolete

AI State Information

NPCs that are sub-classed from ScriptedPawn may be given orders that they will attempt to carry out during the course of the game, in the absence of overriding behaviors (attacking, fleeing, etc.). These orders are specified by the Order variable. If an action is to be carried out on a specific actor, the OrderTag variable is used to specify which actor. The NPC will find the nearest actor whose Tag variable is set to the same name as OrderTag. If OrderTag is not set, the player is used as the actor.

The following behaviors may be set using the Order variable:
Idle: Causes the NPC to stand in one place and not react to anything.

Standing: Causes the NPC to stand in one place and react to events normally. Using the OrderTag variable, it is possible to specify a HomeBase actor that the NPC must stand at. The NPC will also face the direction the HomeBase is facing. If the NPC moves for any reason, and eventually reverts to his Standing orders, he will automatically return to the HomeBase; if no HomeBase is specified, he will stand in place.

Dancing: Causes the NPC to stand in one place and dance. This is almost identical to the Standing state, except that the NPC will play a dancing animation and periodically turn (if bHokeyPokey is enabled).

Wandering: Causes the NPC to wander randomly. Using the OrderTag variable, it is possible to specify a HomeBase actor that the NPC must remain close to. The NPC will attempt to stay within visual range of the HomeBase, and he will attempt to remain within the radius specified in the HomeBase’s Extent variable.

Sitting: Causes the NPC to sit down in a chair (any subclass of Seat). If OrderTag is set, the NPC will look for a seat with the specified tag; otherwise, he will simply sit in the nearest seat. If no seats exist (or are otherwise occupied), the NPC will go into a Standing state.

Patrolling: Causes the NPC to follow a chain of patrol points. OrderTag specifies the starting patrol point, and the next patrol point in the chain is determined by the patrol point’s NextPatrol variable. If an NPC arrives at a patrol point, and no further patrol points are in the chain, the NPC will enter a Standing state.

Following: Causes the NPC to follow a specific pawn. The pawn to be followed is specified by the OrderTag variable. The NPC will try to remain close to the pawn at all times, and will run if necessary to catch up. If the pawn being followed gets too close to the NPC, the NPC will eventually back up (to prevent hallway blockage and other problems).

Shadowing: Causes the NPC to follow a specific pawn in circumspect fashion. The pawn to be followed is specified by the OrderTag variable. The NPC will try to keep the pawn in sight at all times, running if necessary. If the pawn is watching the NPC, the NPC will appear to be in a Wandering state. This is the paranoid ”is-he-following-me-or-isn’t-he” state.

WaitingFor: Causes the NPC to wait until he sees a specific pawn. Once the pawn is spotted, the NPC moves toward the pawn and tries to touch him (usually to initiate conversation). The pawn is specified via the OrderTag variable.

The following states are for scripting purposes only. They should not be specified using the Orders variable.
GoingTo: Causes the NPC to walk towards a specific actor, specified by the ubiquitous OrderTag variable.

RunningTo: Same as GoingTo, but causes the NPC to run to the actor designated by OrderTag.

Property Name	Field Name	Type	Description
ScriptedPawn	BHighlight	bool	True – NPC can be highlighted to select
ScriptedPawn	BImportant	bool	True – The death of this NPC will be flagged so that the NPC will be deleted from any subsequent maps they are to appear on.
ScriptedPawn	BInvincible	bool	True – The NPC cannot be killed.
ScriptedPawn	BInWorld	bool	True – The NPC is created in the world. NPC can be made to appear later via scripting if this is set to False to start with.
ScriptedPawn	ClotPeriod	float	
ScriptedPawn	WalkSound	sound	Sound file for the walking noise of the NPC.

Sounds

The fields within this property will be different depending on the class of the NPC or Pawn. All of the fields are for sound files and correspond to everything form Death sounds to Target Lost sounds.

Triggers

Several new trigger types were added to increase the functionality available to the designers. By default all of the triggers have collisions set to be on. There are several other properties and fields that can be adjusted to increase the functionality of all triggers. Most triggers can be chained together into a mess of convoluted logic. The ones used by the DE group are listed below.

Property Name	Field Name	Type	Description
Collision	BCollideActors	bool	True – Actors entering collision radius of trigger will set it off.
Collision	CollisionHeight	float	Collision height of trigger in game units
Collision	CollisionRadius	float	Collision radius of trigger in game units
Trigger	BInitiallyActive	bool	True – Trigger is active when first encountered
Trigger	BTriggerOnceOnly	bool	True – Trigger can be set off once only, then it becomes inert
Trigger	ClassProximityType	class<*>	Any class type of actor can be placed here. None is the Default. This will only be checked if TriggerType = TT_ClassProximity. If TriggerType is set, then this will be the type of class that can only fire the trigger.
Trigger	ReTriggerDelay	float	Time delay between triggering. Useful to keep from firing the trigger too many times and causing mover difficulties
Trigger	TriggerType	menu	Selection of types: **TT_PlayerProximity** (Player sets off trigger – Default). **TT_PawnProximity** (any pawn sets it off. If you want a specific pawn use class proximity and that Pawns class type). **TT_ClassProximity** (specific class type *ClassProximityType* will set off trigger). **TT_AnyProximity** (any actor class or pawn will set off trigger). **TT_Shoot** (Weapon fire sets off trigger).
Events	Event	string	Tag name of Actor, Pawn or other class item (movers, other

			triggers, etc.) to activate by this trigger.
Events	Tag	string	ID name of trigger. Used as the activation name if called by other triggers or actors.

AllianceTriggers

These allow the designer to have the alliance of NPCs to be altered under certain conditions. These can be linked to other triggers allowing for conditional checks to be made prior to shifting the alliances

Property Name	Field Name	Type	Description
Alliance Trigger	Alliance	string	Name of alliance to change effected NPC(s) to
Alliance Trigger	Alliances	varies [8]	Alliance Level (float) – hostility level of NPC to this alliance (-1.0.1) AllianceName (string) – Alliance name bPermenant (bool) – Can alliance level for this alliance be altered through other parameters.
Events	Event	string	Tag name of NPC(s) to be effected by trigger
Events	Tag	string	Name of trigger

BeamTrigger

This is an object producing a blue laser like beam in the direction indicated by the arrow in the overhead or side views of the editor. By standardization, DE used this beam exclusively for activating movers.

Property Name	Field Name	Type	Description
BeamTrigger	bison	bool	True – Beam is on when first encountered
Events	Event	string	Tag name of mover(s) to be effected by trigger
Events	Tag	string	Name of trigger

ConversationTrigger

Used to initiate a conversation when nearing an NPC or event area and the normal conversation radius on the NPC will not be the best solution.

Property Name	Field Name	Type	Description
ConversationTrigger	bCheckFalse	bool	True – Trigger if CheckFlag = false
ConversationTrigger	BindName	string	Name of NPC that is to do the talking (This is their bind name too)
ConversationTrigger	CheckFlag	string	Name of flag to check before running conversation. If left as Default (none), no flag check will occur
ConversationTrigger	conversationTag	string	Conversation name that is to be played

DamageTrigger

Used in areas that you want to damage the player. Deus Ex used these in environmental areas (radiation, poisonous water, etc).

Property Name	Field Name	Type	Description
DamageTrigger	bConstantDamage	bool	True – Damage will occur as long as player (or whatever) is within the radius of the trigger.
DamageTrigger	DamageAmount	float	Damage delivered to target on each time interval
DamageTrigger	damageInterval	float	Time in seconds between damage
DamageTrigger	DamageType	string	One of the viable damage types needs to be used here. Valid types are: Shot, Exploded, TearGas, PoisonGas, HalonGas, Radiation, Flamed (catches player on fire), Burned, Shocked, NanoVirus, EMP, Drowned, Stunned.

DataLinkTrigger

This trigger is used to activate the Data Link conversations.

Property Name	Field Name	Type	Description
DataLinkTrigger	bCheckFalse	bool	True – Trigger if CheckFlag = false
DataLinkTrigger	CheckFlag	string	Name of flag to check before running conversation. If left as Default (none), no flag check will occur
DataLinkTrigger	datalinkTag	string	Name of the data link conversation to be played.

FlagTrigger

This was most often used in conjunction with other triggers to provide conditional checks on game state or to set flags indicating a task was accomplished or area reached. Very useful and was used quite a bit in DE. These triggers can be daisy-chained effectively.

Property Name	Field Name	Type	Description
Events	Event	string	Tag name of trigger(s) to be effected by this flag trigger
Events	Tag	string	Name of flag trigger
FlagTrigger	bSetFlag	bool	True - Will set flag of *FlagName* to *flagValue*
FlagTrigger	bTrigger	bool	True - Will trigger *Event* name in the Events property
FlagTrigger	bWhileStandingOn	bool	True – Untrigger flag when player or actor leaves collision of radius
FlagTrigger	flagExpiration	int	Mission number that flag will expire (be deleted) on. Defaults to –1, meaning after current mission.
FlagTrigger	FlagName	string	Name of the flag
FlagTrigger	flagValue	bool	Value to set flag or check flag for

GoalCompleteTrigger

Goals are initiated through conversations but at several times the goal needs to be indicated as complete once the task is accomplished. The goal complete trigger will erase the goal form the data vault.

Property Name	Field Name	Type	Description
GoalCompleteTrigger	goalName	string	Name of the goal to clear. A check of the conversation file will be necessary to get the name assigned to the goal.

LaserTrigger

This is an object that produces a blue laser like beam in the direction indicated by the arrow in the overhead or side views of the editor. By standardization, DE used this beam for initiating alarms or auto turrets.

Property Name	Field Name	Type	Description
LaserTrigger	bIsOn	bool	True – laser trigger is on when first encountered
LaserTrigger	bNoAlarm	bool	True – laser does not sound an alarm.
Events	Event	string	Tag name of event(s) to be effected by trigger
Events	Tag	string	Name of trigger

LogicTrigger

One of the more complex triggers. This one will check for an event from two other triggers and will do a logical (Boolean) operation on their input and then will put out the appropriate trigger based on the outcome. Great for such things as seeing if two switches have been turned on before releasing the robots.

Property Name	Field Name	Type	Description
LogicTrigger	inGroup1	string	Input one. *Group* field name under *Object* property of first trigger feeding logic Trigger
LogicTrigger	inGroup2	string	Input two. *Group* field name under *Object* property of second trigger feeding logic Trigger
LogicTrigger	Not	bool	True - Invert output of logic trigger (True becomes False, False becomes True)

LogicTrigger	OneShot	bool	True – Logic trigger will fire only once
LogicTrigger	Op	menu	Selection of viable Boolean operations: GATE_AND, GATE_OR, GATE_XOR

Note that the *Events* property, Event name of both triggers that feed the LogicTrigger must match the *Events* property, Tag name of the LogicTrigger.

OrdersTrigger

These triggers allow the designer to issue new orders to an NPC while the game is in operation based on other events or conditions.

Property Name	Field Name	Type	Description
OrdersTrigger	Orders	name	Order type to give to the NPC. See section above on AI orders.
OrdersTrigger	ordersTag	name	Tag name to reference order to. Could be a patrol point name or another NPC.
Events	Event	string	Tag name of NPC(s) to be effected by trigger
Events	Tag	string	Name of trigger

SenseTrigger

Verifies that an item is in the player or pawn's inventory and if it is, then the trigger is activated.

Property Name	Field Name	Type	Description
SenseTrigger	KeyNeeded	class<Inventory>	Item required to be in inventory.
Events	Event	string	Tag name of NPC(s) to be effected by trigger
Events	Tag	string	Name of trigger

SequenceTrigger

Used to trigger elevator movers or multiMovers (see above)

Property Name	Field Name	Type	Description
SequenceTrigger	SeqNum	int	Used with ElevatorMover to go to a certain key frame. Used with the MultiMover to go to a certain set of key frames
Events	Event	string	Tag name of mover(s) to be effected by trigger
Events	Tag	string	Name of trigger

Shake Trigger

This creates a visual effect of shaking the users screen, but it also effects the play environment too. With a bit of extra scripting it can be made to move objects on the screen and rain debris down on the player (As during the conclusion of the Ship mission in Deus Ex).

Property Name	Field Name	Type	Description
ShakeTrigger	shakeRollMagnitude	float	Amplitude of roll displacement (Default = 1024.0)
ShakeTrigger	shakeTime	float	Duration of shake in seconds. (Default =1)
ShakeTrigger	shakeVertMagnitude	float	Amplitude of vertical displacement (Default = 16.0)

SkillAwardTrigger

Once certain events occurred, tasks were accomplished or areas reached the game can award skill points to the player . This can also be done through conversations.

Property Name	Field Name	Type	Description
SkillAwardTrigger	awardMessage	string	Message displayed on screen when award is given
SkillAwardTrigger	skillPointsAdded	int	value of skill award.

Untrigger

Some things like Laser triggers and Beam triggers must be untriggered to turn them off. The Untrigger provides fast functionality to do this. The only fields that need to be affected are the event fields.

Property Name	Field Name	Type	Description
Events	Event	string	Tag name of item(s) to be effected by trigger
Events	Tag	string	Name of trigger