

Documentation

for

Wargame/RUSE Modding suite

by enohka

[Introduction](#)

[Supported file formats](#)

[edata file format \(*.dat; *.pk\)](#)

[Ndfbin file format \(*.ndfbin, *.kdt\)](#)

[trad file format \(*.dic\)](#)

[tgx file format \(*.tgx\)](#)

[Getting started](#)

[Requirements](#)

[Start the program](#)

[Dictionary Editor](#)

[Editing textures](#)

[Requirements](#)

[Export textures](#)

[Ndfbin Editor](#)

[Filter](#)

[Follow object references](#)

Introduction

This document is meant to give an overview about the Wargame: Modding Suite and it's built-in features.

Supported file formats

The Modding Suite supports following file Formats:

edata file format (*.dat; *.*pk)

The edata file format is a *package* of files in a virtual folder structure. The entire gamedata is packed into such files.

The gamedata packages can be found at "*\\Steam\\SteamApps\\common\\Wargame Airland Battle\\Data\\wargame\\PC*".

Ndfbin file format (*.ndfbin, *.kdt)

Ndfbin files contain "compiled" bytecode of NDF scripts. NDF stands for *Neurospaces Description Format*.

NDF is the primary scripting language of Eugen System, the majority of game logic and object definitions are written in it.

trad file format (*.dic)

Trad files are simple dictionary files. Every entry in this dictionary has a *Token*. These tokens are used for the game [localisation](#).

tgx file format (*.tgx)

Tgx files are texture images. This format is comparable to usual *.dds files ([DirectDraw Surface](#), texture files for DirectX). The Modding Suite handles tgx files by converting them into *.dds files and vice versa.

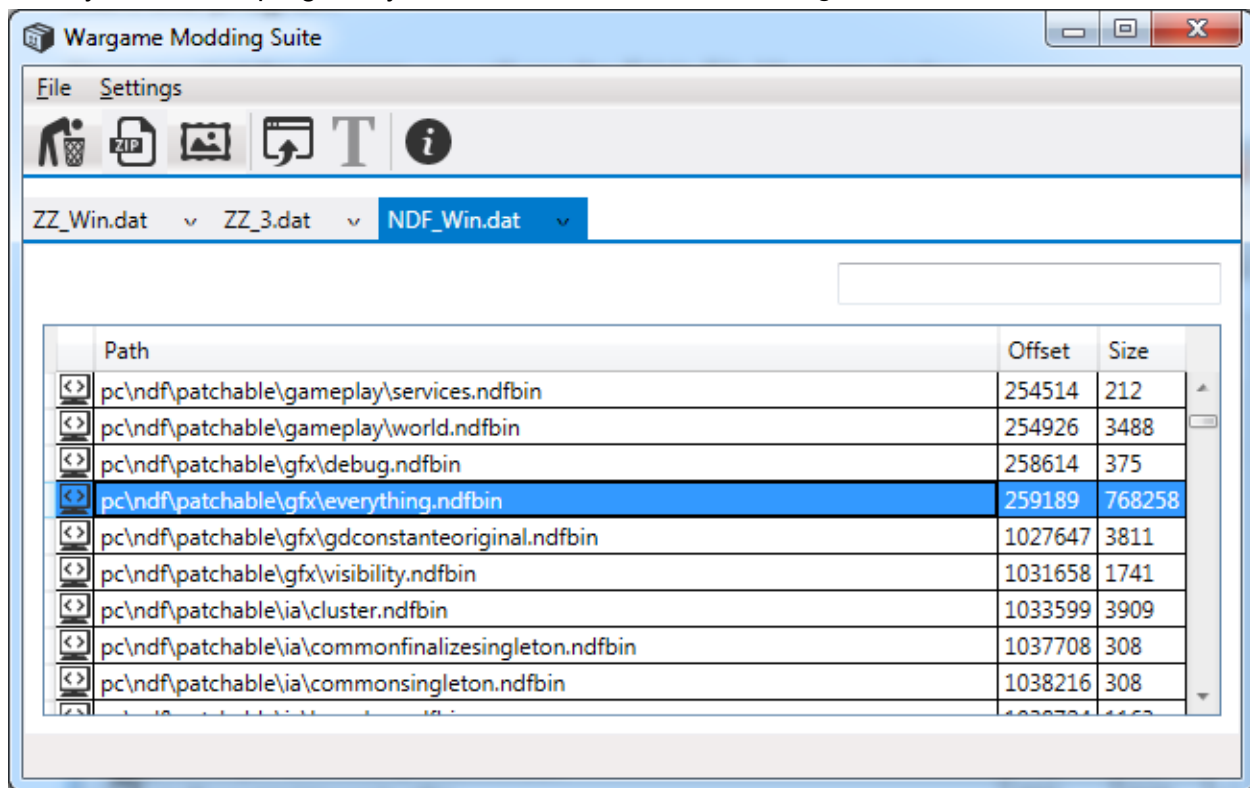
Getting started

Requirements

The Modding Suite requires the *.NET 4.5 framework* which you can [download here](#). Additionally I recommend you to run the software as *Administrator*.

Start the program

Once you start the program, you will see the *Edata File Manager* window.



Edata File Manager

To open a gamedata file, select File->Open and go to your Wargame folder (*Steam\SteamApps\common\Wargame Airland Battle\Data\wargame\PC*). In there, you can see a lot of folders named with a number. These numbers are the game versions. For the first trials, I recommend you to choose a older game version for instance "2100001491". For Wargame: Airland Battle, most of the game logic is stored in the "NDF_win.dat" file.

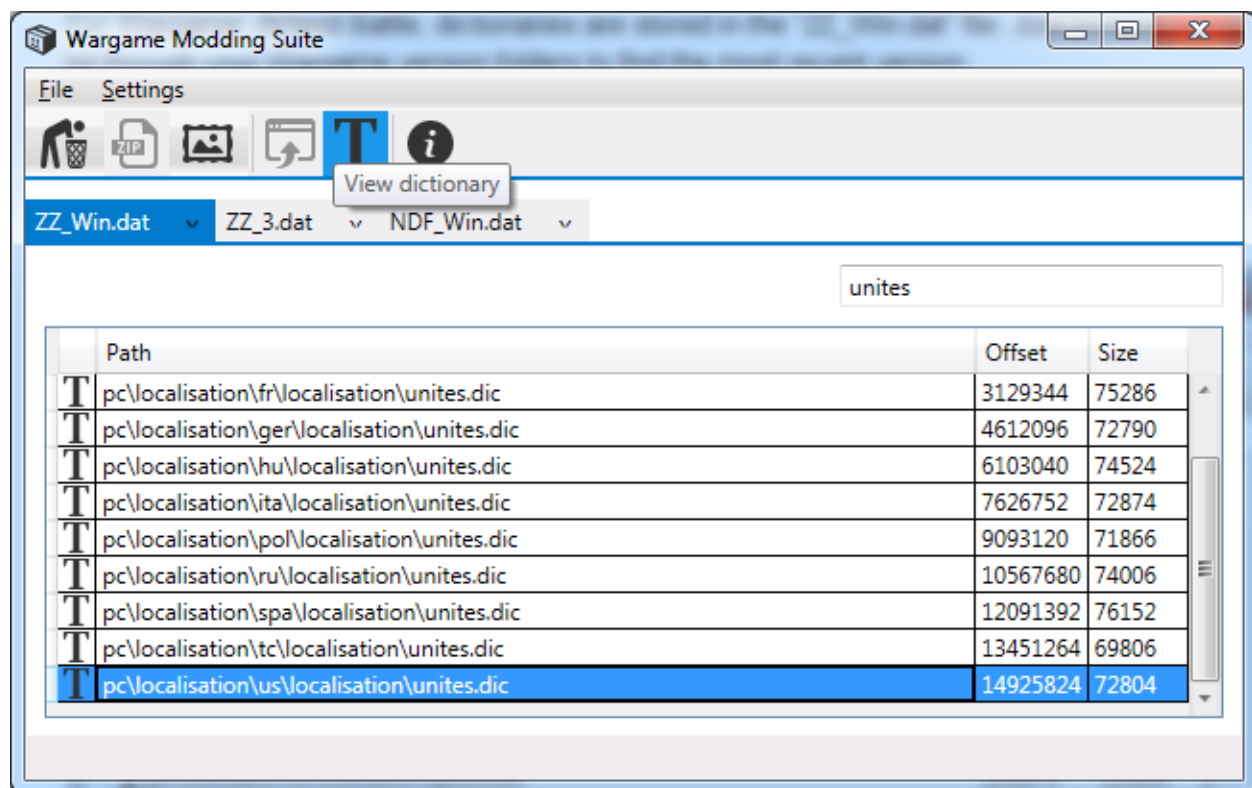
Once you've opened it, a new tab appears, you can see all files that the package contains in the list below. Use the textbox above the list to search for specific files inside the edata package.

Dictionary Editor

The dictionary viewer makes it possible to resolve localisation tokens. It makes it a lot easier to find weapons and units in the Ndfbin editor.

For Wargame: Airland Battle, dictionaries are stored in the “ZZ_Win.dat” file. Just browse a little bit through your Wargame version folders to find the most recent version.

Dictionaries are marked with the **T** Icon in the Edata File Manager and can be viewed by pressing this button in the Toolbar.

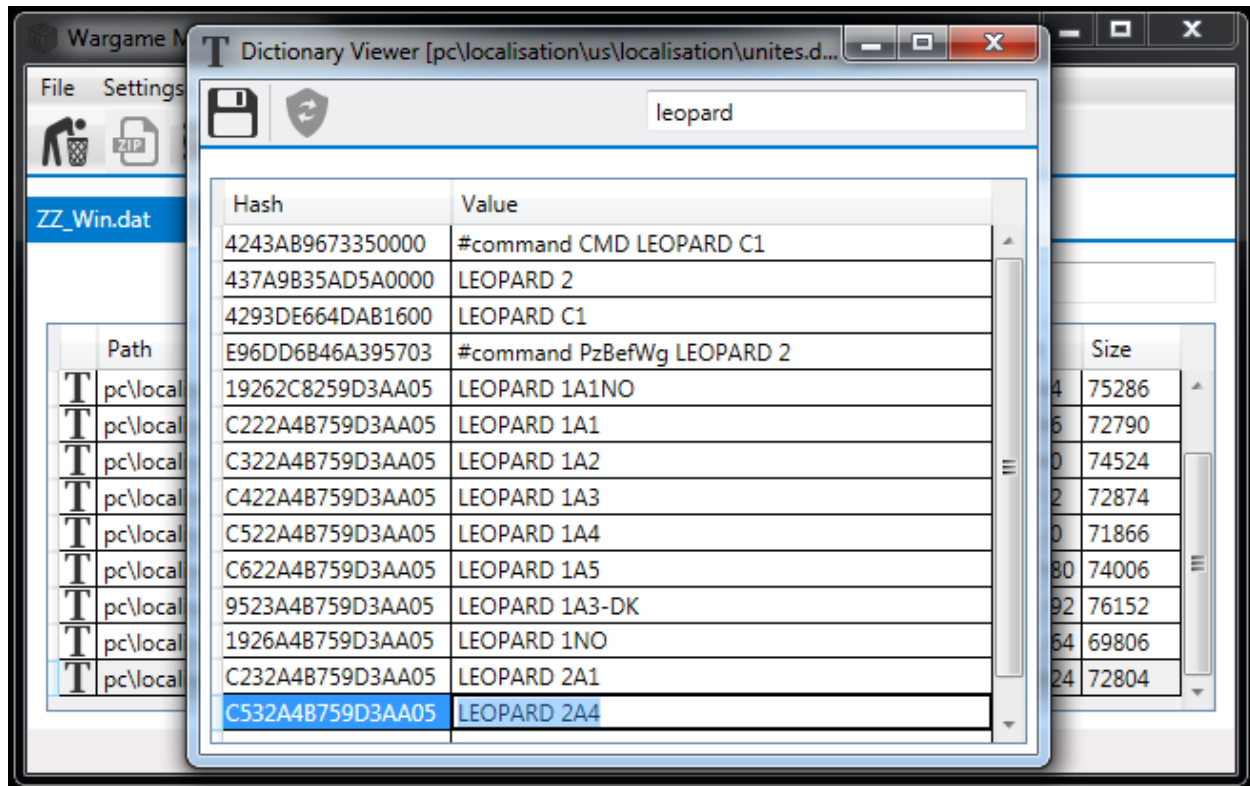


Open a dictionary

The unit and weapon names are stored in the “pc\<countrycode>\localisation\unites.dic” file.


Once you have opened a dictionary file, a dialog appears in which you can search, edit and add localisation entries. You can copy these values (Hash and Value) by double-clicking in the field and pressing **Ctrl + C**.

Hint: These values can be used for the built in [filter](#).



Dictionary editor

To add a new dictionary entry go to the bottom of the list and doubleclick into the *Value field*. You can then enter your desired value. Press the *Return key* to commit the changes.

After you have created a new entry with a value, you need to create a localization hash. Do this by selecting the newly created entry and pressing the  *button* in the toolbar.

Press the *save button* after you're done with your changes.

Editing textures


Requirements

To edit Textures you need any software which is able to handle *.dds files. I recommend Adobe Photoshop with the [NVIDIA Texture Tools for Adobe Photoshop](#) Plug-in.

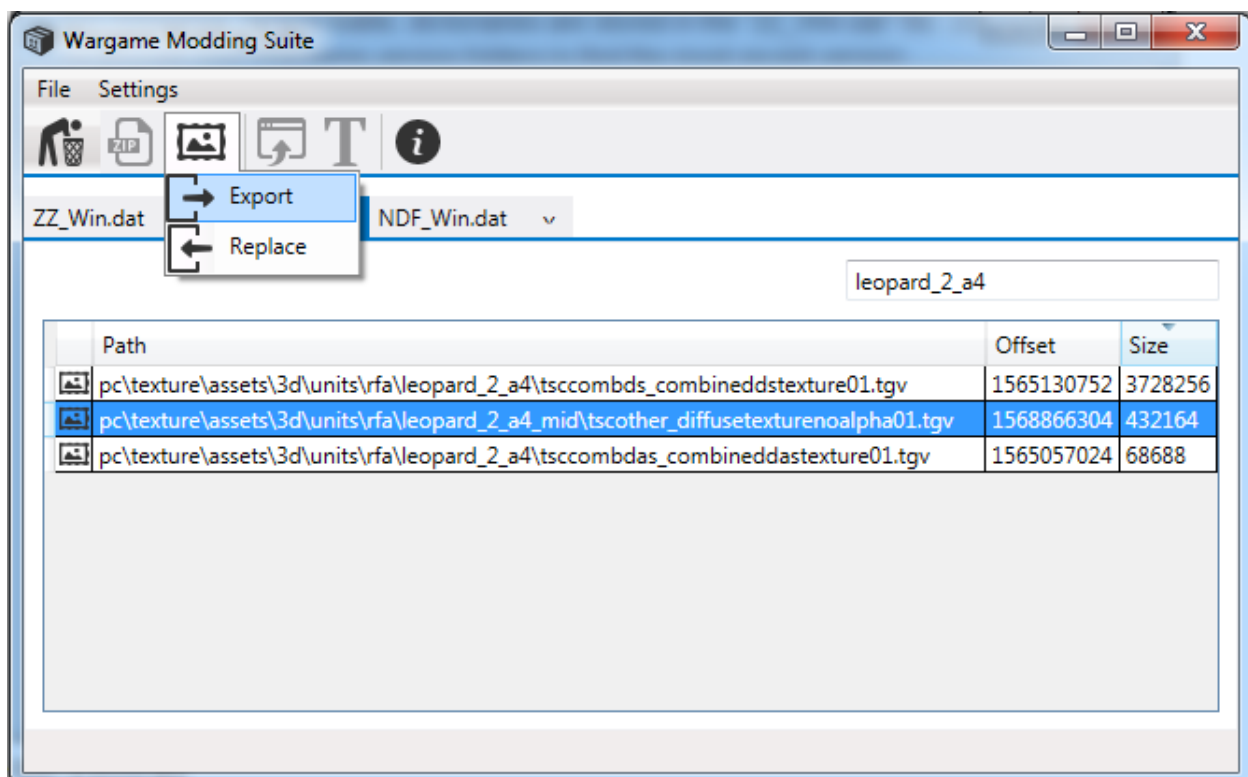
A valid export path must be setted. (Settings->change export path). Keep in mind that you need to have sufficient user rights to save at this location.

Export textures

For Wargame: Airland Battle, textures are mostly located inside the ZZ_3.dat file. Remember to create a backup of this one somewhere. This file is about 3 gigabytes big and you have to redownload it entirely if something goes wrong or you just want to revert your changes.

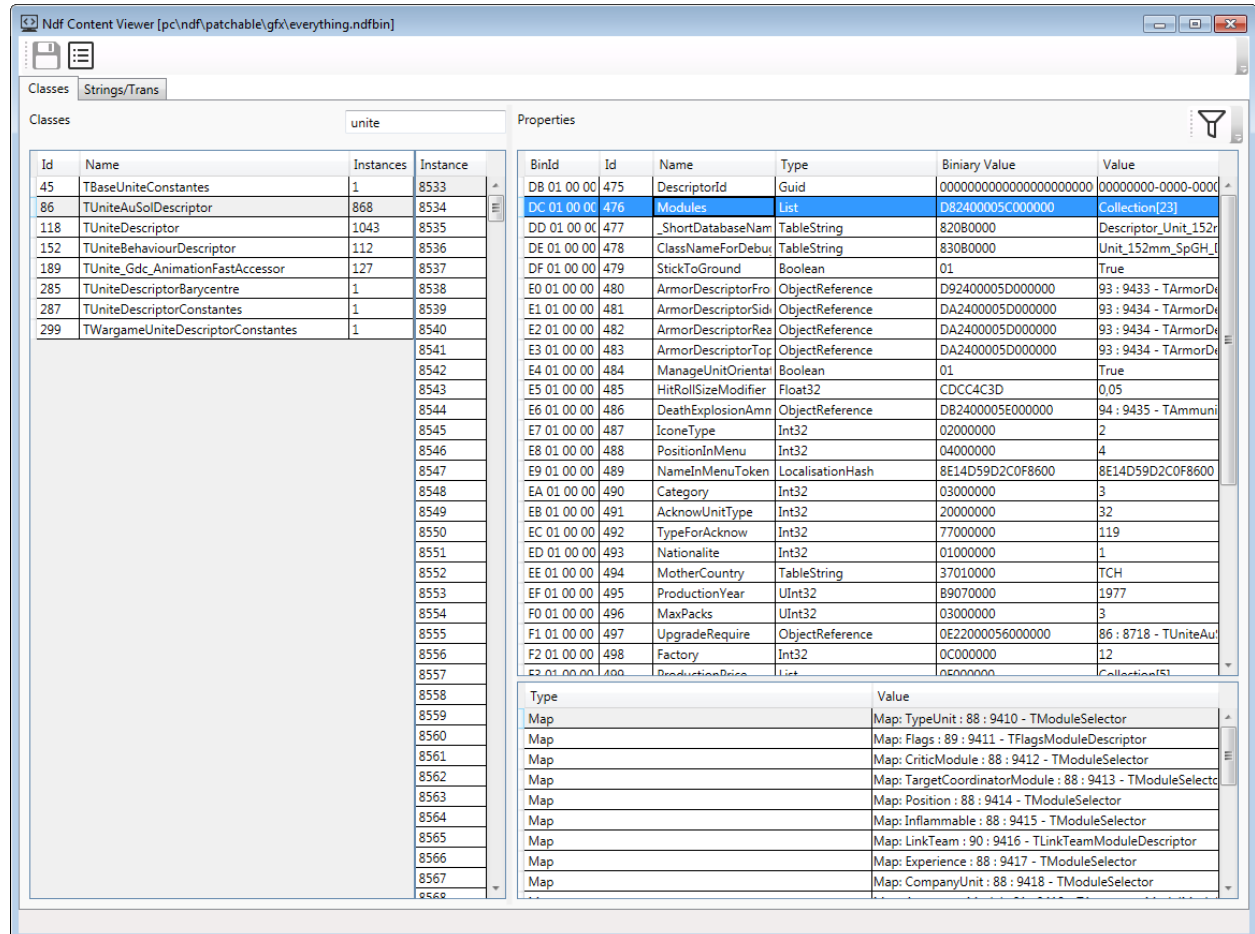
Select the texture to export in the Edata File Manager and press on the  button. Now choose "Export". The file will be converted into a *.dds and is located at your *export path*.

You can now edit the texture with for instance Adobe Photoshop and reimport it using the *replace command*. Note that this may take a while since the whole ZZ_3 file has to be rebuild.



Ndfbin Editor

This feature is probably the largest and most complicated inside the modding suite. It needs to be better documented, sorry.



The screenshot shows the Ndf Content Viewer application. The 'Classes' tab is active, displaying a list of classes. The 'Strings/Trans' tab is also visible, showing a list of strings. The 'Classes' tab has columns for Id, Name, Instances, and Instance. The 'Strings/Trans' tab has columns for BinId, Id, Name, Type, Binary Value, and Value.

Id	Name	Instances	Instance
45	TBaseUniteConstantes	1	8533
86	TUniteAuSolDescriptor	868	8534
118	TUniteDescriptor	1043	8535
152	TUniteBehaviourDescriptor	112	8536
189	TUnite_Gdc_AnimationFastAccessor	127	8537
285	TUniteDescriptorBarycentre	1	8538
287	TUniteDescriptorConstantes	1	8539
299	TWargameUniteDescriptorConstantes	1	8540
			8541
			8542
			8543
			8544
			8545
			8546
			8547
			8548
			8549
			8550
			8551
			8552
			8553
			8554
			8555
			8556
			8557
			8558
			8559
			8560
			8561
			8562
			8563
			8564
			8565
			8566
			8567
			8568

BinId	Id	Name	Type	Binary Value	Value
DB 01 00 00	475	DescriptorId	Guid	00000000-0000-0000-0000-00000000	00000000-0000-0000-0000-00000000
DC 01 00 00	476	Modules	List	D82400005C000000	Collection[23]
DD 01 00 00	477	_ShortDatabaseName	TableString	82080000	Descriptor_Unit_152r
DE 01 00 00	478	ClassNameForDebug	TableString	83080000	Unit_152mm_SpGH_1
DF 01 00 00	479	StickToGround	Boolean	01	True
E0 01 00 00	480	ArmorDescriptorFront	ObjectReference	D92400005D000000	93 : 9433 - TArmorDe
E1 01 00 00	481	ArmorDescriptorSide	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDe
E2 01 00 00	482	ArmorDescriptorRear	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDe
E3 01 00 00	483	ArmorDescriptorTop	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDe
E4 01 00 00	484	ManageUnitOrientation	Boolean	01	True
E5 01 00 00	485	HitRollSizeModifier	Float32	CDCC4C3D	0.05
E6 01 00 00	486	DeathExplosionAmn	ObjectReference	DB2400005E000000	94 : 9435 - TAmmoni
E7 01 00 00	487	IconType	Int32	02000000	2
E8 01 00 00	488	PositionInMenu	Int32	04000000	4
E9 01 00 00	489	NameInMenuToken	LocalisationHash	8E14D59D2C0F8600	8E14D59D2C0F8600
EA 01 00 00	490	Category	Int32	03000000	3
EB 01 00 00	491	AcknowUnitType	Int32	20000000	32
EC 01 00 00	492	TypeForAcknow	Int32	77000000	119
ED 01 00 00	493	Nationalite	Int32	01000000	1
EE 01 00 00	494	MotherCountry	TableString	37010000	TCH
EF 01 00 00	495	ProductionYear	UInt32	B9070000	1977
FO 01 00 00	496	MaxPacks	UInt32	03000000	3
F1 01 00 00	497	UpgradeRequire	ObjectReference	0E22000056000000	86 : 8718 - TUniteAu
F2 01 00 00	498	Factory	Int32	0C000000	12
F3 01 00 00	499	ProductionPrice	List	05000000	Collection[5]

Type	Value
Map	Map: TypeUnit : 88 : 9410 - TModuleSelector
Map	Map: Flags : 89 : 9411 - TFlagsModuleDescriptor
Map	Map: CriticModule : 88 : 9412 - TModuleSelector
Map	Map: TargetCoordinatorModule : 88 : 9413 - TModuleSelecto
Map	Map: Position : 88 : 9414 - TModuleSelector
Map	Map: Inflammable : 88 : 9415 - TModuleSelector
Map	Map: LinkTeam : 90 : 9416 - TLinkTeamModuleDescriptor
Map	Map: Experience : 88 : 9417 - TModuleSelector
Map	Map: CompanyUnit : 88 : 9418 - TModuleSelector

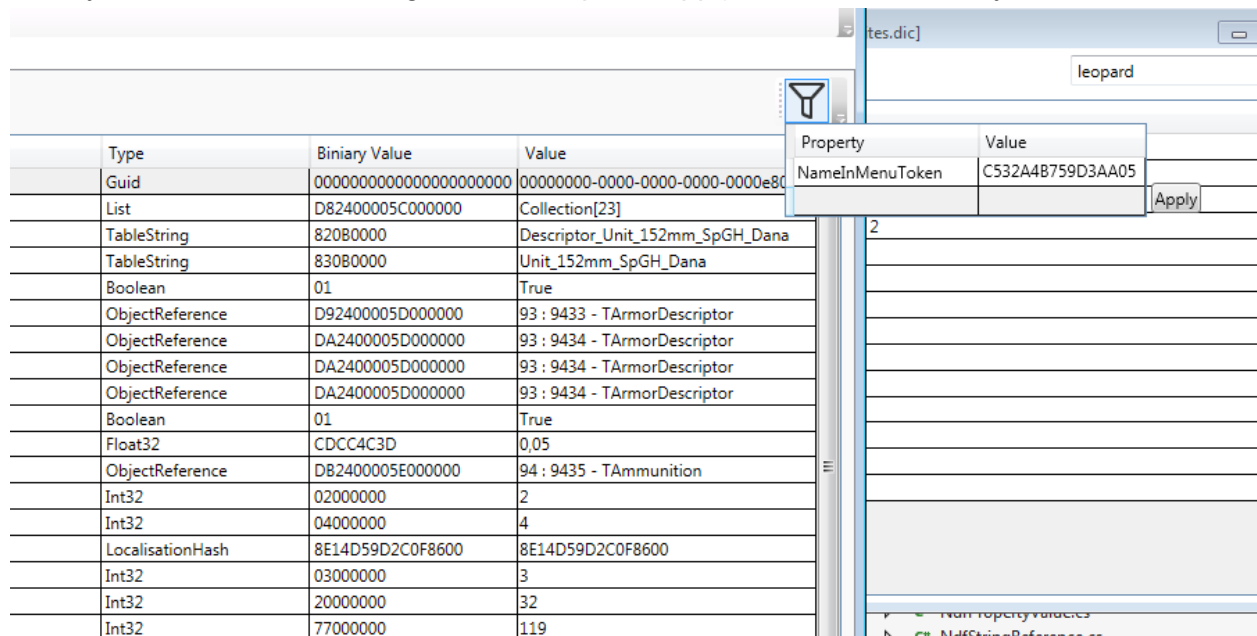
The Ndfbin Editor

Filter

The filter function allows you to search for object instances by having an unlimited amount of conditions.

To add a condition, you have to select any property the current class owns and a value which fits the objects you are looking for. After you've defined your condition press enter to submit it.

Once you are done with adding conditions, press *Apply* to use the filter on your current class.



The screenshot shows a software interface with a table of object instances. A filter dialog box is open, allowing the user to add conditions. The dialog has a search bar at the top with the text "leopard". Below the search bar, there is a table with two columns: "Property" and "Value". The first row in this table shows "NameInMenuToken" as the property and "C532A4B759D3AA05" as the value. To the right of the "Value" column is an "Apply" button. The background table has four columns: "Type", "Binary Value", and "Value". It lists various object types such as Guid, List, TableString, Boolean, ObjectReference, Float32, Int32, and LocalisationHash, along with their corresponding binary and decimal values.

Type	Binary Value	Value
Guid	00000000000000000000000000000000	00000000-0000-0000-0000-0000e800
List	D82400005C000000	Collection[23]
TableString	82080000	Descriptor_Unit_152mm_SpGH_Dana
TableString	83080000	Unit_152mm_SpGH_Dana
Boolean	01	True
ObjectReference	D92400005D000000	93 : 9433 - TArmorDescriptor
ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
Boolean	01	True
Float32	CDCC4C3D	0,05
ObjectReference	D82400005E000000	94 : 9435 - Tammunition
Int32	02000000	2
Int32	04000000	4
LocalisationHash	8E14D59D2C0F8600	8E14D59D2C0F8600
Int32	03000000	3
Int32	20000000	32
Int32	77000000	119

Adding filter conditions

Follow object references

Everytime you see the type "ObjectReference" you can follow this reference by doubleclicking on the row. A new window displaying the referenced instance will appear.

DE 01 00 00	478	ClassNameForDebug	TableString	830B0000	Unit_152mm_SpGH_Dana
DF 01 00 00	479	StickToGround	Boolean	01	True
E0 01 00 00	480	ArmorDescriptorFront	ObjectReference	D92400005D000000	93 : 9433 - TArmorDescriptor
E1 01 00 00	481	ArmorDescriptorSides	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
E2 01 00 00	482	ArmorDescriptorRear	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
E3 01 00 00	483	ArmorDescriptorTop	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
E4 01 00 00	484	ManageUnitOrientation	Boolean	01	True
E5 01 00 00	485	HitRollSizeModifier	Float32	CDCC4C3D	0,05

BinId	Id	Name	Type	Binary Value	Value
DB 01 00 00	475	DescriptorId	Guid	00000000000000000000000000000000	00000000-0000-0000-0000-0000e8030000
DC 01 00 00	476	Modules	List	D82400005C000000	Collection[23]
DD 01 00 00	477	_ShortDatabaseName	TableString	820B0000	Descriptor_Unit_152mm_SpGH_Dana
DE 01 00 00	478	ClassNameForDebug	TableString	830B0000	Unit_152mm_SpGH_Dana
DF 01 00 00	479	StickToGround	Boolean	01	True
E0 01 00 00	480	ArmorDescriptorFront	ObjectReference	D92400005D000000	93 : 9433 - TArmorDescriptor
E1 01 00 00	481	ArmorDescriptorSides	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
E2 01 00 00	482	ArmorDescriptorRear	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
E3 01 00 00	483	ArmorDescriptorTop	ObjectReference	DA2400005D000000	93 : 9434 - TArmorDescriptor
E4 01 00 00	484	ManageUnitOrientation	Boolean	01	True
E5 01 00 00	485	HitRollSizeModifier	Float32	CDCC4C3D	0,05
E6 01 00 00	486	ProductionCost	UInt32	03000000	3
E7 01 00 00	487	MaxPacks	UInt32	03000000	3
E8 01 00 00	488	UpgradeRequire	ObjectReference	0E22000056000000	86 : 8718 - TUnitAuSolDescriptor

TArmorDescriptor

TArmorDescriptor : 9434

BinId	Id	Name	Type	Binary Value	Value
29 02 00 00	553	Name	LocalisationHash	46F95DDC52620000	46F95DDC52620000
2A 02 00 00	554	BaseBlindage	UInt32	05000000	5

Type	Value
------	-------