

Scripting guide for Lineage 2 Adrenaline

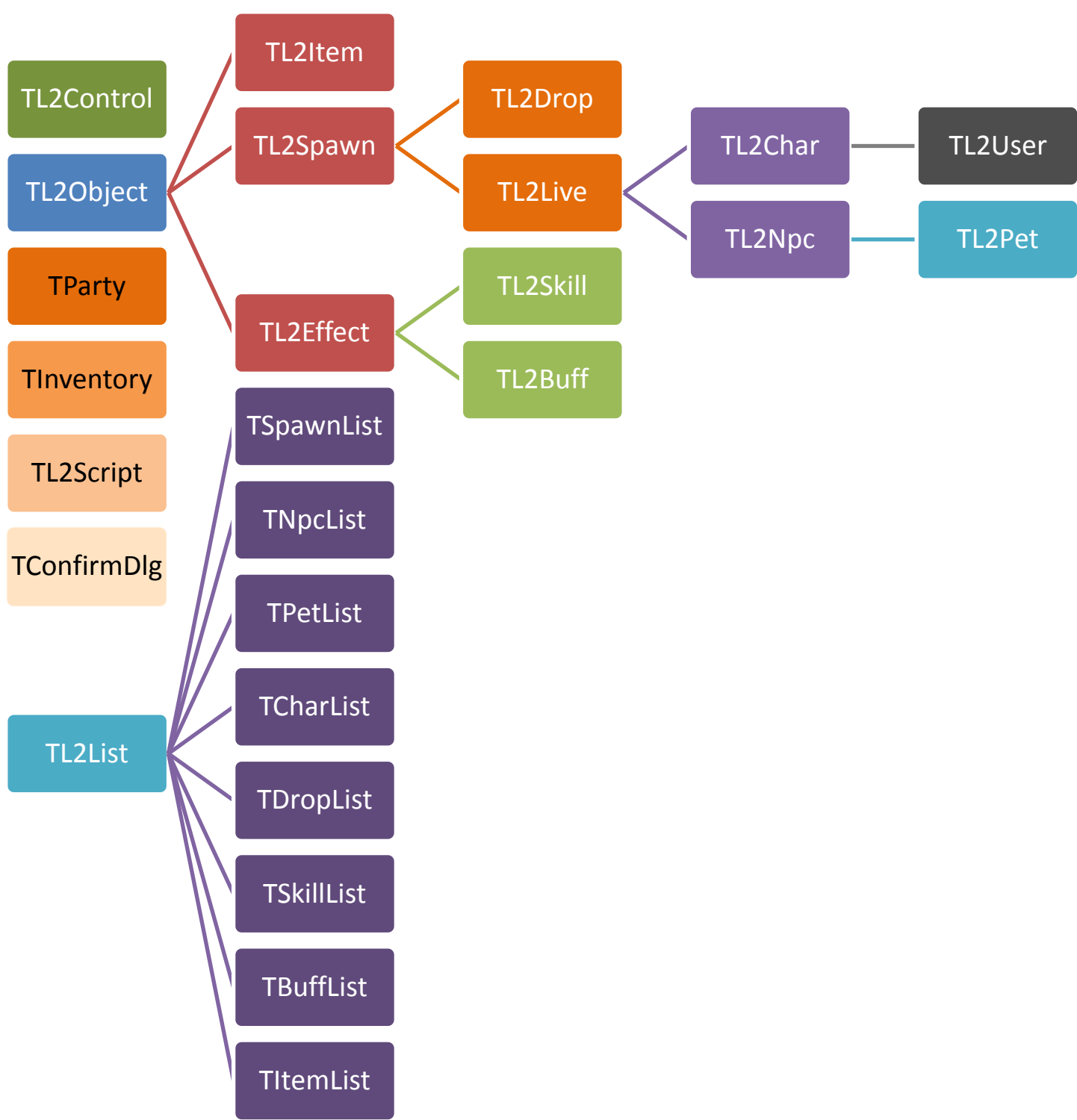
Questions related to program activation (key purchase):
Skype: natherlog

by Novikov
Localization is made
by natherlog

Contents

Class hierarchy.....	2
TL2Control = class	3
TL2Object = class	5
TL2Spawn = class(TL2Object);	5
TL2Drop = class(TL2Spawn);	6
TL2Npc = class(TL2Live);	6
TL2Pet = class(TL2Npc);	6
TL2Char = class(TL2Live);	6
TL2User = class(TL2Char);	6
TL2Effect = class(TL2Object);	6
TL2Buff = class(TL2Effect);	6
TL2Live = class(TL2Spawn);	7
TL2Skill = class(TL2Effect);	8
TL2Item = class(TL2Object);	8
TL2List = class;	8
TSpawnList = class(TL2List);	9
TNpcList = class(TL2List);	9
TPetList = class(TL2List);	9
TCharList = class(TL2List);	9
TDropList = class(TL2List);	9
TSkillList = class(TL2List);	9
TBuffList = class(TL2List);	9
TItemList = class(TL2List);	9
TParty = class;	10
TInventory = class;	10
TL2Script = class;	10
TConfirmDlg = class.....	10
Functions:	10
ChatMessage	10
Enumeration types	11
Examples.....	13

Class hierarchy



TL2Control = class

TL2Control = class	General class of bot control. Available in the scripts functions by name Engine. Can be received from another account engine with the help of GetControl(const Name: string): TL2Control;
Assist(const Name: string): Boolean;	Assists the target (e.g Engine.Assist(name))
Attack(TimeOut: Cardinal = 2000; Ctrl: Boolean = false): Boolean;	Attack. TimeOut - delay (ms); Ctrl - force attack
AutoSoulShot(const Name: string; Active: Boolean): Boolean;	Enable/disable automatical soulshot use. Name - soulshot name; Active - on/off
AutoTarget(Range: Cardinal = 2000; ZRange: Cardinal = 300; NotBusy: Boolean = true): Boolean;	Target autoselection in radius Range. NotBusy - selects only "free target" (not targeted by anyone). Considers map zones and objects added to the IgnoreList
BlinkWindow(GameWindow: Boolean): Boolean;	Blinks on the taskbar. If GameWindow - true then blinks game window, otherwise blinks bot window
BotWindow: Cardinal;	Get HWND of lineage 2 window (Bot window)
ByPassToServer(const S : String) : Boolean;	
CancelTarget: Boolean;	Cancel the target
ClearIgnore;	Clear the ignore list
ClearZone;	Clear all zones on the map
CrystalItem(ID : Cardinal) : Boolean;	Crystallize the item
ConfirmDialog(Accept: Boolean): Boolean;	Confirm any dialogue in the game (Yes/No)
ConfirmDlg : TConfirmDlg;	Receive class TConfirmDlg
DestroyItem(const Name: string; Count: Cardinal): Boolean;	Destroy item with Name at the rate of Count
DismissParty(const Name: string): Boolean;	Expell player with Name from group
DismissPet: Boolean;	Unsummon pet (if any)
DismissSum: Boolean;	Unsummon summon (if any)
Dispel(const Name: string): Boolean;	Dispell buff with the Name
DlgOpen: Boolean;	Start dialogue with NPC
DlgSel(const Txt: string): Boolean; overload;	Select string by Text (e.g Engine.DlgSel('Quest.'))
DlgSel(Index: integer): Boolean; overload;	Select string by Index (e.g Engine.DlgSel(1))
DlgText: string;	Contains full text of the current dialogue
DMoveTo(x, y, z : Integer) : Boolean;	Move to the set point without waiting for a finish
DUseSkill(id : Cardinal; ctrl, Shift : Boolean) : Boolean;	Use skill by ID without cooldown check and MP amount.
Entry(var Param): Boolean;	Function activation in the script of another account. Activating function should look like - function OnEntry(var Param): Boolean; Param - any transmissive parameter.
EnterText(const Txt: string): Boolean;	Press Enter, write Txt, press Enter.
Equipped(const Name: string): Integer;	Equipment check
FaceControl(ID: Integer; Active: Boolean): Boolean;	On/off interface button.
FindEnemy(var Enemy: TL2Live; Obj: TL2Live; Range: Cardinal = 2000; ZRange: Cardinal = 300): Boolean;	Search an "enemy" for the object Obj in the set Radius (regarding to object Obj). If function result - true, then found "enemy" will be written to Enemy variable.
GameClose: Boolean;	Close the game (L2 client)
GameStart(CharIndex: Integer): Boolean;	Start the game by CharIndex (characters should be on the character selection screen)
GameWindow : Cardinal;	Get HWND of lineage 2 window (Game Window)
GetUser: TL2User;	Getting user object (TL2User) from another account
GetParty: TParty;	
GetSkillList: TSkillList;	
GetInventory: TInventory;	
GetDropList: TDropList;	
GetNpcList: TNpcList;	
GetCharList: TCharList;	
GetPetList: TPetList;	
GoHome(ResType: TRestartType = rtTown) : Boolean;	Return character back to village after death
Ignore(Obj: TL2Spawn);	Add an object to the ignore list. AutoTarget and AutoPick up methods skip such objects.

InviteParty(const Name: string; Loot: TLootType = ldLooter): Boolean;	Invite player by Name. Loot - looting rules
InZone(X, Y, Z: Integer): Boolean; overload ;	If there is the point in zone
InZone(Obj: TL2Spawn): Boolean; overload ;	If there is an object in zone
IsBusy(Obj: TL2Npc): Boolean;	Check object if it is targeted or not by another players
IsDay: Boolean;	Day in the game
JoinParty(Join: Boolean): Boolean;	Accept/refuse party invitation
LearnSkill(ID: Cardinal): Boolean;	Learn skill by ID. Character should be near the skill master in the HighFive or older chronicles.
LeaveParty: Boolean;	
LoadZone(const Name: string): Boolean;	Load zone on the map from the file.
LoadConfig(const Name: string): Boolean;	Load the confing by Name from the Settings folder (by default); can be full path as well.
MakeItem(Index : Cardinal) : Boolean;	Craft an item by index (Index - list of crafting recipes)
MoveItem(const Name: string; Count: Cardinal; ToPet: Boolean): Boolean;	Move/take the item with Name from pet in number Count
MoveTo(ToX: integer; ToY: integer; ToZ: integer): Boolean; overload ;	Move to point
MoveTo(Obj: TL2Spawn; Dist: Integer): Boolean; overload ;	Move to the object Obj at a distance of Dist
MoveToTarget(Dist: Integer = -100): Boolean;	Move to the target (Dist should be negative number) (e.g Engine.MoveTo(-75));
MSG(Who, What : String; Color : Integer);	Write system message in the bot window with the different colors.
NpcExchange(ID: Cardinal; Count: Cardinal): Boolean;	NPC exchange. ID - id of item which should be received, Count - in what amount.
NpcTrade(Sell: Boolean; items: array of Cardinal): Boolean;	Trade with NPC. Sell - purchase/sell. Items - massive of items for purchase sell. Massive should consist of pairs (ID, Count) and should be 2-fold
OpenQuestion: Boolean;	Click question mark (needed for specific quests)
Pickup(Range: Cardinal = 250; ZRange: Cardinal = 150; OnlyMy: Boolean = false; Pet: Boolean = false): Integer; overload ;	AutoPickup of all drop in the radius Range. OnlyMy - only mine. Pet - pick up by pet
Pickup(Obj: TL2Drop; Pet: Boolean = false): Boolean; overload ;	Pick up object Obj. Pet - pick up by pet
QuestStatus(QuestID: Cardinal; Step: Integer): Boolean;	Quest status check. Step - quest step
Restart: Boolean;	Go back to character selection screen (character should be out of fight)
Say(const Text: string; ChatType: Cardinal = 0; const Nick: string = ''): Boolean;	Say in chat
SendMail(const Recipient: string; const Theme: string; const Content: string; Items: array of Cardinal; Price: Cardinal = 0): Boolean;	Mail sending. Recipient - addressee; Theme - theme; Content - content; Items - list (massive) of attached items (ID/amount); Price - price (if you set any price it will be "Payment request" otherwise "Regular mail"). If there is no item with set ID, it will be automatically excluded from the mail (sending will not be canceled). If the amount of any item is more than real, it will be automatically changed
ServerTime: Cardinal;	
GameTime: Cardinal;	
SetPartyLeader(const Name: string): Boolean;	Change party leader with the Name (your character should be party leader)
SetTarget(ID: Cardinal): Boolean; overload ;	Set target by ID
SetTarget(const Name: string): Boolean; overload ;	Set target by name
SetTarget(Obj: TL2Live): Boolean; overload ;	Set object Obj as target
Sit: Boolean;	Sit
Stand: Boolean;	Stand
StopCasting : Boolean;	Stop skill cast
Status: TL2Status;	Current account status
Unstuck : Boolean;	Unstuck;

UpdateSkillList: Boolean;	Open skill list (for Interlude servers)
UseAction(ID: Cardinal; Ctrl: Boolean = false; Shift: Boolean = false): Boolean;	Use ingame action
UseItem(ID: Cardinal; Pet: Boolean = false): Boolean; overload;	Use item by ID. Pet - used by pet
UseItem(const Name: string; Pet: Boolean = false): Boolean; overload;	Use item by Name. Pet - used by pet
UseItem(Obj: TL2Item; Pet: Boolean = false): Boolean; overload;	Use item Obj. Pet - used by pet
UseKey(Key: Word; Ctrl: Boolean = False; Shift: Boolean = False): Boolean; overload;	Press key. Hold Shift, Ctrl
UseKey(const Key: string; Ctrl: Boolean = False; Shift: Boolean = False): Boolean; overload;	
UseSkill(ID: Cardinal; Ctrl: Boolean = false; Shift: Boolean = false): Boolean; overload;	Use skill by ID
UseSkill(const Name: string; Ctrl: Boolean = false; Shift: Boolean = false): Boolean; overload;	Use skill by name
WaitAction(Actions: TL2Actions; var P1; var P2; TimeOut: Cardinal = INFINITE): TL2Action;	Wait for event or group of events

TL2Object = class

TL2Object = class	Basic class for all ingame objects
ID: Cardinal;	Object ID
Name: String;	Object Name
OID: Cardinal;	Unique identifier for any object in the game
Valid: Boolean;	Check object for validity in the game
SetVar(Value: Cardinal);	Set variable for Object
GetVar: Cardinal;	Get value of variable
L2Class: TL2Class;	Find out the class which current object belongs to

TL2Spawn = class(TL2Object);

TL2Spawn class(TL2Object) ;	All objects around the character
DistTo(X: Integer; Y: Integer; Z: Integer): Cardinal; overload;	Return the distance to the set point
DistTo(Obj: TL2Spawn): Cardinal; overload;	Return the distance to the object Obj
InRange(X: Integer; Y: Integer; Z: Integer; Range: Cardinal; ZRange: Cardinal = 250): Boolean;	Check if the point is in the Range (regarding to the Object) in the set radius
InZone: Boolean;	Check if the object is in the zone (hunting zone)
SpawnTime: Cardinal;	Object spawn time
X: Integer;	Coordinates of the object
Y: Integer;	Coordinates of the object
Z: Integer;	Coordinates of the object

TL2Drop = class(TL2Spawn);

TL2Drop = class(TL2Spawn) ;	Drop in the game
Count: int64;	Amount
IsMy: Boolean;	Check if drop is mine (mine - dropped by our character, pet or our party)
Stackable: Boolean;	Stackable or no

TL2Npc = class(TL2Live);

TL2Npc = class(TL2Live) ;	Basic class for all NPC
IsPet: Boolean;	Pet or no
PetType: Cardinal;	Pet type (summon or no)

TL2Pet = class(TL2Npc);

TL2Pet = class(TL2Npc) ;	Class describing our pets/summons
Fed: Cardinal;	Food (only in %)

TL2Char = class(TL2Live);

TL2Char = class(TL2Live) ;	Basic class for all characters
CP : Cardinal;	
CurCP : Cardinal;	
MaxCP : Cardinal;	
Hero : Boolean;	
Noble : Boolean;	
ClassID : Cardinal;	
MainClass: Cardinal;	
MountType: Byte;	Mount type
StoreType: Byte;	
Sex : Cardinal;	0 male 1 female
Race : Cardinal;	
CubicCount: Cardinal;	
Recom: Cardinal;	
Premium : Boolean;	

TL2User = class(TL2Char);

TL2User = class(TL2Char) ;	Class describing our character
CanCryst: Boolean;	Can our character crystallize items or no
Charges: Cardinal;	Charges for Duelists
WeightPenalty: Cardinal;	
WeapPenalty: Cardinal;	
ArmorPenalty: Cardinal;	
DeathPenalty: Cardinal;	
Souls: Cardinal;	Souls for Kamaels

TL2Effect = class(TL2Object);

TL2Effect = class(TL2Object) ;	Basic class for all magic skills
Level: Cardinal;	Skill level
EndTime: Cardinal;	Cooldown

TL2Buff = class(TL2Effect);

TL2Buff = class(TL2Effect) ;	Class describing buffs
------------------------------	------------------------

TL2Live = class(TL2Spawn);

TL2Live = class(TL2Spawn) ;	Basic class of "alive" objects in the game (player, npc, pet etc)
AbnormalID : Cardinal;	ID of visual effect.
Abnormals : TBuffList;	available in the GoD+ chronicles or above
Ally: string;	Alliance name
AllyID: Cardinal;	Alliance ID of the object
Attackable: Boolean;	Attackable (not force attack)
AtkOID: Cardinal;	OID of the object which is attacking
AtkTime: Cardinal;	The time object has started to attack
Bufs: TBuffList;	Bufs of the object (available for our character, party members and pets)
Cast: TL2Cast;	Skill of the object which is casted in the current moment. Actual if Cast.EndTime > 0, otherwise object is not currently casting.
Clan: string;	Clan name
ClanID: Cardinal;	Clan ID of the object
CurHP: Cardinal;	Current amount of Health Points
CurMP: Cardinal;	Current amount of Mana Points
Dead: boolean;	Dead or alive
Dropped: Boolean;	Object has dropped or hasn't dropped an item (Dead parameter must be True)
Exp: Int64;	Experience
EXP2: Int64;	
Fishing: Integer;	
Fly: Boolean;	If the object has flying or non-flying status
HP: Cardinal;	Current amount of HP (%)
InCombat: Boolean;	In chaotic stance (in fight)
IsMember: Boolean;	Object is member of party
Karma: Integer;	Karma (in GoD+ chronicles can be positive as well)
Level: Byte;	
Load: Cardinal;	Weight (%) (available for our character and pet)
MaxHP: Cardinal;	Maximal HP amount
MaxMP: Cardinal;	
MP: Cardinal;	Current amount of MP (%)
MyAtkTime: Cardinal;	When I attacked the object.
PK: Boolean;	Player Killer
PvP: Boolean;	The object is in PvP mode
Running: Boolean;	The object is moving or jogging
Sitting: Boolean;	Sitting?
SP: Cardinal;	Skill Points (SP)
Speed: Double;	
Sweepable: Boolean;	Sweepable
Target: TL2Live;	The target of an object
Team: Byte;	For PvP servers (blue/red type of monsters), as well as "champion" monsters
TeleportDist: Cardinal;	The distance of the last teleport
TeleportTime: Cardinal;	The time of the last teleport
Title: string;	Object title
ToX: Integer;	The coordinates where object moved
ToY: Integer;	
ToZ: Integer;	

TL2Skill = class(TL2Effect);

TL2Skill = class(TL2Effect) ;	Skill class
Disabled: Boolean;	Skill is not available
Enchanted: Boolean;	This is Enchanted, a member of class TL2Skill.
Passive: Boolean;	Skill is passive

TL2Item = class(TL2Object);

TL2Item = class(TL2Object) ;	Inventory items class
Count: Int64;	Amount (if stackable)
Equipped: Boolean;	Equipped or no
EnchantLevel: Word;	This is EnchantLevel, a member of class TL2Item.
ItemType : Cardinal;	0 weapon; 1 armor; 2 jewelry; 5 resources and all other items
Grade: Cardinal;	
GradeName: string;	('NG', 'D', 'C', 'B', 'A', 'S', 'S80', 'S84', 'R', 'R95', 'R99');

TL2List = class;

TL2List = class;	All the lists of the objects in bot are inherited from this class, lists: buffs, NPC, chars, drops, items.
ByID(ID: Cardinal; var Obj): Boolean;	Search for an object in the list by ID. If the object is found, it is moved in the Obj variable.
ByName(const Name: string; var Obj): Boolean;	Search for an object in the list by name. If the object is found, it is moved in the Obj variable.
Count: integer;	Amount of the objects in the list
Items(Index: integer): TL2Object;	Allows you to access the object in the list by index

TSpawnList = class(TL2List);

TSpawnList = class(TL2List);	List of all the objects that have coordinates.
Items(Index: integer): TL2Spawn;	Allows you to access the object in the list by index

TNpcList = class(TL2List);

TNpcList = class(TL2List);	List of the surrounding NPC. In the scripts available as NpcList
Items(Index: integer): TL2Npc;	Allows you to access the object in the list by index

TPetList = class(TL2List);

TPetList = class(TL2List);	List of pets. In the scripts available as PetList
Items(Index: integer): TL2Pet;	Allows you to access the object in the list by index

TCharList = class(TL2List);

TCharList = class(TL2List);	List of all surrounding players. In the scripts available as CharList
Items(Index: integer): TL2Char;	Allows you to access the object in the list by index

TDropList = class(TL2List);

TDropList = class(TL2List);	List of all drop around us. In the scripts available as DropList
Items(Index: integer): TL2Drop;	Allows you to access the object in the list by index

TSkillList = class(TL2List);

TSkilllList = class(TL2List);	List of the current skills of the character. In the scripts available as SkillList
Items(Index: integer): TL2Skill;	Allows you to access the object in the list by index

TBuffList = class(TL2List);

TBuffList = class(TL2List);	Object buffs list.
Items(Index: integer): TL2Buff;	Allows you to access the object in the list by index

TItemList = class(TL2List);

TItemList = class(TL2List);	Inventory list.
Items(Index: integer): TL2Item;	Allows you to access the object in the list by index

TParty = class;

TParty = class;	Party class. In the scripts available as Party.
Pets: TNpcList;	List of pets in the party
Chars: TCharList;	List of characters in the party
LootType: TLootType;	Looting rules (types)
Leader: TL2Char;	Party leader

TInventory = class;

TInventory = class;	Inventory class. In the scripts available as Inventory.
Pet: TItemList;	Pet inventory
User: TItemList;	Character inventory
Quest: TItemList;	Character inventory (quest)

TL2Script = class;

TL2Script = class;	Script class
MainProc(Proc: Pointer);	Call up the procedure of the main thread of the program. Proc - pointer on the thread procedure; Parameter - any transferable parameter (not necessary)
NewThread(Proc: TThreadFunc; Parameter: Pointer = nil);	Create new script thread. Proc - pointer on the thread procedure (procedure MyTHread(Prm.Pointer)) Parameter - any transferable parameter (not necessary)
Path: String;	Full path to the directory of script
Resume;	Resume script
Suspend;	Pause script

TConfirmDlg = class

TConfirmDlg = class	Dialogue class
MsgID : Cardinal;	
ReqID : Cardinal;	
Sender: string;	
EndTime: Cardinal;	
Valid : Boolean;	

Функции:

Функции:	Неочевидные
OnFree;	Only forending the script. The body of the procedure must be created by yourself.
PlaySound(const FileName: string; Loop: Boolean);	Проигрывает звуковой файл в формате wave (.wav). Loop - зациклить воспроизведение.
StopSound;	Stop sound
Delay(ms: Cardinal): Boolean;	Delay for script (ms)
ExePath: string;	Returns the path of root folder of the bot
GetControl(const Name: string): TL2Control;	Get the Engine object of another character. Name - the name of character.
BotLoginID: Cardinal;	Receive unique ID (hash) based on bot-login.

ChatMessage

ChatMessage	Chat managing object
ChatMessage.unread: boolean;	Unread message
ChatMessage.sender: string;	Sender of chat message
ChatMessage.text: string;	Text of sender
ChatMessage.chattype: TMessageType;	Type of sent message

Перечисляемые типы.

[illegible]

```
TRestartType = (  
    rtTown,  
    rtClanHoll,  
    rtCastle,  
    rtFort,  
    rtFlags  
);
```

```
TMessageType = (  
    mtSystem,  
    mtAll,  
    mtPrivate,  
    mtParty,  
    mtClan,  
    mtFriend,  
    mtShout  
);
```

```
TL2Class = (  
    lcError,  
    lcDrop,  
    lcNpc,  
    lcPet,  
    lcChar,  
    lcUser,  
    lcBuff,  
    lcSkill,  
    lcItem  
);
```

ChatMessage

```
var
  Obj: TL2Live;
begin
  while True do
    begin
      if ChatMessage.Unread and (ChatMessage.Time < 3000) then
        begin
          if CharList.Byname(ChatMessage.sender, Obj) and not Obj.IsMember then
            if Obj.InZone then
              begin
                print(ChatMessage.sender);
                print(ChatMessage.text);
                print(ChatMessage.Time);
                print(ChatMessage.ChatType);
              end;
            end;
          Delay(111);
        end;
      end.
    end.
```

Type in the system tab (window) of bot, the message of player, which is in zone and not your party member.

Function TL2Live.AbnormalID

```
Function IsUD(actor : TL2Live) : Boolean;
begin
  Result := actor.AbnormalId and $8000000 = $8000000;
end;

//Target in UD?
begin
  if isUD(User.target) then
    print('Yes, in UD');
  end.
```

Check if target has UD or no. Value \$8000000 - it is one of bites of variable for AbnormalID, designative the effect of UD.

Some other effects:

```
bleeding = $1;
poison = $2;
redcircle = $4;
ice = $8;
```

function TL2Control.GetSkillList

For current script, **TL2Control** is available by name Engine as well as list of skills available by variable **SkillList**. From another **TL2Control** skill list can be received with the help of **GetSkillList**.

```
Function GetSkill(const Control : String; const ID : Cardinal) : TL2Skill;
var
  NovObj : TL2Skill;
  NovEngine : TL2Control;
begin
  NovEngine := GetControl(control);
  if Assigned(NovEngine) and NovEngine.GetSkillList.ByID(ID, NovObj) then
    Result := NovObj;
  end;

begin
  if Assigned(GetSkill('MyBuffer',1255)) then
    GetControl('MyBuffer').UseSkill(1255) //Recall
  Else
    Engine.UseItem(736); //Юзает coe
end.
```

If our buffer with nickname 'MyBuffer' has skill Party Recall, then we use it. In other case, we use Scroll of Escape (SoE).

function TL2Control.GameWindow

What do we need this game window HWND for?
For example, you can use WinApi functions in the script.

```
function SetForegroundWindow(hwnd: integer) : Boolean; stdcall;
external 'user32.dll';

begin
    While Engine.Delay(5000) then
        if User.dead then
            SetForegroundWindow(Engine.GameWindow);
        end.
end.
```

Scroll up game window, if character is dead.

procedure TL2Control.MSG(Who, What : String; Color : Integer);

If you want to paint system chat, you can use this method. The color can be set by any values which are situated below:

```
clBlack = 0; //Black
clMaroon = 128; //Maroon
clGreen = 32768; //Green
clOlive = 32896; //Olive
clNavy = 8388608; //Navy
clPurple = 8388736; //Purple
clTeal = 8421376; //Teal
clGray = 8421504; //Gray
clSilver = 12632256; //Silver
clRed = 255; //Red
clLime = 65280; //Lime
clYellow = 65535; //Yellow
clBlue = 16711680; //Blue
clFuchsia = 16711935; Fuchsia
clAqua = 16776960; //Aqua
clWhite = 16777215; //White
```

TL2Live.Teleport...

Be familiar with information of any object, when teleported and how far including information about myself.

```
begin
    While Engine.Delay(1000) do
        if (GettickCount - User.TeleportTime < 5000) and (User.TeleportDist < 2000) and
(User.TeleportDist > 10) then //Teleport was done 5 seconds ago
            begin
                print('Teleported!');
                PlaySound(exepath + '\sounds\' + 'dc' + '.wav', False);
            end;
        end.
end.
```

If teleported, play sound dc.wav from the bot folder (bot/sounds)
