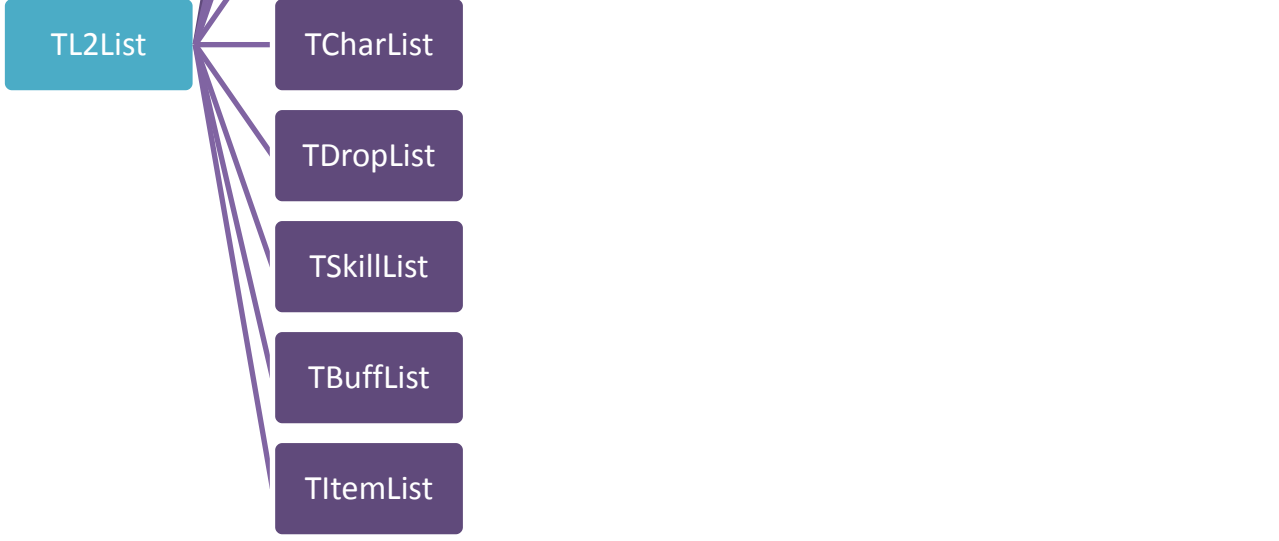
Справочник по скриптам для бота под Lineage 2 Adrenaline

by Novikov

По вопросам активации программы:
ICQ: 292934301

Оглавление

Иерархия классов



Описание классов

TL2Control = class

TL2Control = class	**Основной класс управления ботом. В скриптах доступен по имени Engine. Можно получить для другого окна GetControl(const Name: string): TL2Control;**
`Assist(const Name: string): Boolean;`	Делает своей целью, цель указанного игрока или NPC
`Attack(TimeOut: Cardinal = 2000; Ctrl: Boolean = false): Boolean;`	Атака. TimeOut - задержка (мс); Ctrl - принудительная атака
`AuctionBuyItem(Item: TL2AucItem): Boolean;`	купить с аукциона
`AuctionCancelItem(Item: TL2AucItem): Boolean;`	снимет предмет с продажи
`AuctionGetMySales: Boolean;`	получить список своих лотов продажи
`AuctionSearch(const Name: string; Grade: Integer = -1; PageID: Integer = 0): Boolean;`	Поиск по аукциону. Grade: 0 - NG; 1 - D,.., 10 - R99.
`AuctionSellItem(Item: TL2Item; Count, Price, Days: Cardinal; CustomName: string = ''): Boolean;`	выставить на продажу, Days: 1, 3, 5, 7
`AutoSoulShot(const Name: string; Active: Boolean): Boolean;`	Включает/выключает на шотах автоматический режим. Name - название шотов; Active - вкл/откл
`AutoTarget(Range: Cardinal = 2000; ZRange: Cardinal = 300; NotBusy: Boolean = true): Boolean;`	Авто-подбор цели в радиусе Range. NotBusy - выбирать только "свободную" цель. (учитывает зоны на карте и объекты добавленные в IgnorList)
`BlinkWindow(GameWindow: Boolean): Boolean;`	Мигает окном на панели задач. Если GameWindow - true то мигает окном игры, иначе окном бота
`BotWindow: Cardinal;`	Получить HWND окна с ботом.
`ByPassToServer(const S : String) : Boolean;`	Отправить на сервер ByPass
`CancelTarget: Boolean;`	Сбрасывает таргет
`ClearIgnore;`	Очищает список игнора
`ClearMail: Boolean;`	Очистить почту.
`ClearZone;`	Очистка всех зон на карте.
`ConfirmDialog(Accept: Boolean): Boolean;`	Отвечает на запросы Да/Нет
`ConfirmDlg : TConfirmDlg;`	Получить класс TConfirmDlg;
`CreateRoom(Text: string; LevelStart, LevelEnd: Integer): Boolean;`	Создать комнату группы.
`CloseRoom: Boolean;`	Закрыть комнату группы.
`CrystalItem(ID : Cardinal) : Boolean;`	Кристализовать итем.
`Delay(const time : Cardinal) : Boolean;`	Паузы для выполнения, в отличии от delay глобального, возвращает True по истечению паузы
`DestroyItem(const Name: string; Count: Cardinal): Boolean;`	Уничтожает предмет с названием Name в количесвте Count
`DismissParty(const Name: string): Boolean;`	Исключает игрока с именем Name из группы
`DismissPet: Boolean;`	Отозывает пета (если есть)
`DismissSum: Boolean;`	Отозывает самона (если есть)
`Dispel(const Name: string): Boolean;`	Снимает с вашего персонажа баф с названием Name
`DlgOpen: Boolean;`	Начинает диалог с NPC
`DlgSel(const Txt: string; const TimeOut : Integer = 1000): Boolean; overload;`	Выбирает при диалоге строку Txt
`DlgSel(Index: integer; const TimeOut : Integer = 1000): Boolean; overload;`	Выбирает при диалоге строку с порядковым номером Index. TimeOut время в мс сколько ожидать диалога.
`DlgText: string;`	Содержит полный текст текущего диалога
`DMoveTo(x, y, z : Integer) : Boolean;`	Двигаться в указанную точку без ожидания завершения.
`DUseSkill(id : Cardinal; ctrl, Shift : Boolean) : Boolean;`	Использовать скил с указанным id без проверки на откат, количество мп.
`EnterText(const Txt: string): Boolean;`	Нажать Enter, написать Txt, нажать Enter.
`Entry(var Param): Boolean;`	Вызов функции в скрипте другого аккаунта. Вызываемая ф-ция должна иметь вид - function OnEntry(var Param): Boolean; Param - любой передаваемый параметр.
`Equipped(const Name: string): Integer;`	Проверка экипировки
`FaceControl(ID: Integer; Active: Boolean): Boolean;`	Вкл/откл клавишу интерфеиса.

`FindEnemy(var Enemy: TL2Live; Obj: TL2Live; Range: Cardinal = 2000; ZRange: Cardinal = 300): Boolean;`	Поиск "врага" для объекта Obj в указанном радиусе (относительно объекта Obj). Если результат ф-ции - true, то найденный "враг" будет записан в переменную Enemy.
`FindPath(StartX, StartY, EndX, EndY: Integer; PathList: TList): Boolean;`	Расчитывает путь из Startx,y До Endx,y и помещает точки в PathList подряд.
`GameClose: Boolean;`	Закрывает игру (клиент L2)
`GameStart(CharIndex: Integer = -1): Boolean;`	Заводит персонажа под номером CharIndex в игру (должны находится на панели выбора персонажей), без параметров при -1, зайдет на последнего активного персонажа
`GameTime: Cardinal;`	Текущее игровое время
`GameVersion : Cardinal;`	Game Protocol
`GameWindow : Cardinal;`	Получить HWND окна с игрой.
`GetCharList: TCharList;`	
`GetDailyItems : Boolean;`	Получить все ежедневные предметы.
`GetDropList: TDropList;`	
`GetFaceState(ID: Integer): Boolean;`	Узнать статус кнопки интерфейса. (FaceControl)
`GetInventory: TInventory;`	
`GetMailItems(MaxLoad: Cardinal = 65; MaxCount: Cardinal = 1000): Boolean;`	Получить все письма с ограничениями по максимальное загрузке, и максимальному количеству итемов
`GetNpcList: TNpcList;`	
`GetParty: TParty;`	
`GetPetList: TPetList;`	
`GetSkillList: TSkillList;`	
`GetUser: TL2User;`	Получание объекта User (TL2User) из другого аккаунта.
`GoHome(ResType: TRestartType = rtTown) : Boolean;`	Возвращает персонажа в город после смерти
`Ignore(Obj: TL2Spawn);`	Добавляет объект в список игнгора. Методы AutoTarget и AutoPickup пропускают такие объекты
`InviteParty(const Name: string; Loot: TLootType = ldLooter): Boolean;`	Приглашает в группу игрока с именем Name. Loot - тип распределения дропа в группе
`InZone(Obj: TL2Spawn): Boolean; overload;`	Объект находится в зоне?
`InZone(X, Y, Z: Integer): Boolean; overload;`	Точка находится в зоне?
`IsBusy(Obj: TL2Npc): Boolean;`	Проверяет объект на "занятость" другими игроками
`IsDay: Boolean;`	День в игре?
`JoinParty(Join: Boolean): Boolean;`	Отвечает на приглашение в группу
`LearnSkill(ID: Cardinal): Boolean;`	Учит скил по ID. В HighFive и ниже должны находиться возле тренера
`LeaveParty: Boolean;`	Покидает группу
`LoadConfig(const Name: string): Boolean;`	Загрузка конфига с именем Name. По умолчанию из папки Settings, можно указать полный путь.
`LoadItems(ToWH: Boolean; Items: array of Cardinal): Boolean;`	Работа со складом, ToWH положить или взять с вархауса, Items массив пар, айди итема и число
`LoadZone(const Name: string): Boolean;`	Загрузка зоны на карту из файла.
`MakeItem(Index : Cardinal) : Boolean;`	скрафтить предмет, Index из списка крафта берется.
`MoveItem(const Name: string; Count: Cardinal; ToPet: Boolean): Boolean;`	Передает/забирает предмет с названием Name у пета в количестве Count
`MoveTo(Obj: TL2Spawn; Dist: Integer): Boolean; overload;`	Подойти к объекту Obj на дистанцию Dist
`MoveTo(ToX: integer; ToY: integer; ToZ: integer; const TimeOut : Integer = 8000): Boolean; overload;`	Двигаться в точку. TimeOut время в мс сколько пытаться двигаться в точку
`MoveToTarget(Dist: Integer = -100): Boolean;`	Подойти к цели на дистанцию Dist
`MSG(Who, What : String; Color : Integer);`	Написать системное сообщение в окне бота, различного цвета.
`NpcExchange(ID: Cardinal; Count: Cardinal): Boolean;`	Обмен вещей у NPC. ID - id вещи которую хотим получить, Count - в каком количестве.
`NpcTrade(Sell: Boolean; items: array of Cardinal): Boolean;`	Торговать с Npc. Sell - покупка/продажа. Items - массив предметов для покупки/продажи. Массив должен состоять из пар ID, Count (кратен двум)
`OpenQuestion: Boolean;`	Открывает "знак вопроса" (требуется для некоторых квестов)

`Pickup(Obj: TL2Drop; Pet: Boolean = false): Boolean; overload;`	Подобирает объект Obj. Pet - Подбирает петом
`Pickup(Range: Cardinal = 250; ZRange: Cardinal = 150; OnlyMy: Boolean = false; Pet: Boolean = false): Integer; overload;`	Авто-подбор всего дропа в радиусе Range. OnlyMy - Свой или весь дроп. Pet - Подбирает петом
`PostMessage(Msg: Cardinal; wParam, lParam: Integer): Integer;`	
`QuestStatus(QuestID: Cardinal; Step: Integer): Boolean;`	Проверка выполнен шаг квеста или нет. Step - интересующий шаг квеста.
`Restart: Boolean;`	Выходит на панель выбора персонажей (чар не должен находиться в режиме боя)
`Say(const Text: string; ChatType: Cardinal = 0; const Nick: string = ''): Boolean;`	Написать в чат.
`SendMail(const Recipient: string; const Theme: string; const Content: string; Items: array of Cardinal; Price: Cardinal = 0): Boolean;`	Отправка почты. Recipient - адресат; Theme - тема; Content - содержание; Items - список (массив) прикрепленных вещей (ID/Кол-во); Price - цена (если указана считается "Безопасная сделка" иначе "Простая отправка"). Если вещь с отправляемым ID не найдена у персонажа, она будет автоматически исключена из отправки (отправка не срывается). Если при отправке какой либо вещи указанное кол-во превышает реальное, оно будет автоматически исправлено.
`SendMessage(Msg: Cardinal; wParam, lParam: Integer): Integer;`	ф-ии для отправки сообщений игровому окну л2. Используется для нажатия клавиш клавиатуры / мышки. Подробное описание в гугле. Работает на РуОфе
`ServerTime: Cardinal;`	
`SetPartyLeader(const Name: string): Boolean;`	Передает лидерство в группе игроку с именем Name (ваш персонаж должен быть лидером группы)
`SetTarget(const Name: string): Boolean; overload;`	Взятие цели по имени.
`SetTarget(ID: Cardinal): Boolean; overload;`	Взятие цели по ID.
`SetTarget(Obj: TL2Live): Boolean; overload;`	Взятие объекта Obj в качестве цели.
`Sit: Boolean;`	Сесть
`Stand: Boolean;`	Встать
`Status: TL2Status;`	Текущий статус аккаунта
`StopCasting : Boolean;`	Прервать чтение заклинания.
`Unstuck : Boolean;`	Сделать Unstuck;
`UpdateSkillList: Boolean;`	Открыть скил лист, для IL серверов.
`UseAction(ID: Cardinal; Ctrl: Boolean = false; Shift: Boolean = false): Boolean;`	Использование игровых действий.
`UseItem(const Name: string; Pet: Boolean = false): Boolean; overload;`	Использует предмет по имени. Pet - использует петом
`UseItem(ID: Cardinal; Pet: Boolean = false): Boolean; overload;`	Использует предмет по ID. Pet - использует петом
`UseItem(Obj: TL2Item; Pet: Boolean = false): Boolean; overload;`	Использует предмет Obj. Pet - использует петом
`UseKey(const Key: string; Ctrl: Boolean = False; Shift: Boolean = False): Boolean; overload;`	
`UseKey(Key: Word; Ctrl: Boolean = False; Shift: Boolean = False): Boolean; overload;`	Нажать кнопку. Зажимая Ctrl, Shift
`UseSkill(const Name: string; Ctrl: Boolean = false; Shift: Boolean = false): Boolean; overload;`	Использует скил по имени
`UseSkill(ID: Cardinal; Ctrl: Boolean = false; Shift: Boolean = false): Boolean; overload;`	Использует скил по ID
`WaitAction(Actions: TL2Actions; var P1; var P2;  TimeOut: Cardinal = INFINITE): TL2Action;`	Ожидание события или группы событий

TL2Object = class

TL2Object = class	Базовый класс всех игровых объектов
ID: Cardinal;	ID объекта
Name: String;	Имя объекта
OID: Cardinal;	Уникальный идентификатор для любого объекта в игре.
Valid: Boolean;	Проверка объекта на существование в игре (актуальность)
SetVar(Value: Cardinal);	назначить объекту переменную
GetVar: Cardinal;	получить значение переменной
L2Class: TL2Class;	Узнать класс к которому относится данный объект.

TL2Spawn = class(TL2Object);

TL2Spawn = class(TL2Object);	Все объяекты в не персонажа
DistTo(X: Integer; Y: Integer; Z: Integer): Cardinal; **overload**;	Возвращает дистанцию до заданной точки
DistTo(Obj: TL2Spawn): Cardinal; **overload**;	Возвращает дистанцию до объекта Obj
InRange(X: Integer; Y: Integer; Z: Integer; Range: Cardinal; ZRange: Cardinal = 250): Boolean;	Проверка вхождения точки (относительно объекта) в заданный радиус
InZone: Boolean;	Проверка на вхождение объекта в зону охоты
SpawnTime: Cardinal;	Время появление объекта.
X: Integer;	Координаты объекта
Y: Integer;	Координаты объекта
Z: Integer;	Координаты объекта

TL2Drop = class(TL2Spawn);

TL2Drop = class(TL2Spawn);	Дроп в игре
Count: int64;	Количество
IsMy: Boolean;	Дроп принадлежит нам или нет ("Нам" - если выбил наш чар, пет или члены пати)
Stackable: Boolean;	Стопковый предмет или не может стакаться

TL2Npc = class(TL2Live);

TL2Npc = class(TL2Live);	Базовый класс для всех NPC
IsPet: Boolean;	Пет или нет
PetType: Cardinal;	Тип пета (самон или пет)

TL2Pet = class(TL2Npc);

TL2Pet = class(TL2Npc);	Класс описывающий наших петов/самонов
Fed: Cardinal;	Еда (проценты)

TL2Char = class(TL2Live);

TL2Char = class(TL2Live);	Базовый класс для всех игроков
CP : Cardinal;	
CurCP : Cardinal;	
MaxCP : Cardinal;	
Hero : Boolean;	
Noble : Boolean;	
ClassID : Cardinal;	
MainClass: Cardinal;	
MountType: Byte;	Тип ездового животного
StoreType: Byte;	
Sex : Cardinal;	0 мужик 1 женщина
Race : Cardinal;	
CubicCount: Cardinal;	
Recom: Cardinal;	
Premium : Boolean;	

TL2User = class(TL2Char);

TL2User = class(TL2Char);	**Класс описывающий нашего персонажа**
CanCryst: Boolean;	Может кристализовать предметы наш герой или нет?
Charges: Cardinal;	для гладов зарядки
WeightPenalty: Cardinal;	
WeapPenalty: Cardinal;	
ArmorPenalty: Cardinal;	
DeathPenalty: Cardinal;	
Souls: Cardinal;	для камаэелей души

TL2Effect = class(TL2Object);

TL2Effect = class(TL2Object);	**Базовый класс всех магических эффектов**
Level: Cardinal;	Уровень скила
EndTime: Cardinal;	Время до окончания действия

TL2Buff = class(TL2Effect);

TL2Buff = class(TL2Effect);	Класс описывающий бафы

TL2Live = class(TL2Spawn);

TL2Live = class(TL2Spawn);	Базовый класс "живых" объектов в игре (игрок, нпс, пет и т.д.)
AbnormalID : Cardinal;	айди получившийся из наборов флагов. Примеры ниже.
Abnormals : TBuffList;	Для ГОД+ хроник.
Ally: string;	Имя альянса
AllyID: Cardinal;	ID альянса в который входит объект
Attackable: Boolean;	Свободно атакуемый (без ctrl)
AtkOID: Cardinal;	OID объекта который атакует
AtkTime: Cardinal;	время когда начал атаковать
Buffs: TBuffList;	Бафы объекта (доступны для нашего чара, пета и сопартийцев)
Cast: TL2Effect;	Скил который объект кастует в данный момент. Актуально если Cast.EndTime > 0, иначе объект в данный момент не кастует.
Clan: string;	Имя клана
ClanID: Cardinal;	ID клана в который входит объект
CurHP: Cardinal;	Точное количество жизней
CurMP: Cardinal;	Точное количество маны
Dead: boolean;	Жив или убит
Dropped: Boolean;	Объект выронил предмет или нет (Dead должен быть True)
Exp: Int64;	Опыт
EXP2: Int64;	
Fishing: Integer;	
Fly: Boolean;	This is Fly, a member of class TL2Live.
HP: Cardinal;	Текущее кол-во HP в процентах
InCombat: Boolean;	Объект находится в комбате или нет
IsMember: Boolean;	Является объект членом группы или нет
Karma: Integer;	Карма (начиная с GoD может быть как отрицательной (PK) так и положительной (репутация))
Level: Byte;	Уровень
Load: Cardinal;	Загруженность (проценты) (доступен для нашего чара или петов)
MaxHP: Cardinal;	Максимальное количество ХП
MaxMP: Cardinal;	
Moved : Boolean;	Движется ли объект?
MP: Cardinal;	Текущее кол-во MP в процентах
MyAtkTime: Cardinal;	когда я его атаковал?
PK: Boolean;	Player Killer
PvP: Boolean;	Объект находится в режиме PvP
Running: Boolean;	Объект движется пешком или бегом
Sitting: Boolean;	Сидит?
SP: Cardinal;	Очки SP
Speed: Double;	
Sweepable: Boolean;	Можно свипать?
Target: TL2Live;	Цель объекта
Team: Byte;	для пвп серверов (красное синие подсвечивание), так же мобы "чемпионы"
TeleportDist: Cardinal;	Дистанция последней телепортации
TeleportTime: Cardinal;	Время последней телепортации
Title: string;	Титул объекта
ToX: Integer;	Координаты куда направился объект.
ToY: Integer;	
ToZ: Integer;	

TL2Skill = class(TL2Effect);

TL2Skill = class(TL2Effect);	Класс описывающий скилы
Disabled: Boolean;	Скил не доступен
Enchanted: Boolean;	This is Enchanted, a member of class TL2Skill.
Passive: Boolean;	Скил пасивный

TL2Item = class(TL2Object);

TL2Item = class(TL2Object);	Класс описывающий итемы в инвентаре
Count: Int64;	Количество (если стопка)
Equipped: Boolean;	Вещь надета или нет
EnchantLevel: Word;	This is EnchantLevel, a member of class TL2Item.
ItemType : Cardinal;	0 оружие; 1 броня; 2 бижа; 5 ресурсы и все остальное
Grade: Cardinal;	
GradeName: string;	('NG', 'D', 'C', 'B', 'A', 'S', 'S80', 'S84', 'R', 'R95', 'R99');

TL2AucItem = class(TL2Item)

TL2AucItem = class(TL2Item)	Класс, описывающий предмет аукциона
LotType: Cardinal;	
Seller: string;	Продавец
Price: Int64;	Цена лота за которую сразу можно выкупить предмет. Если равна 0, то выкупить
Days: Cardinal;	
EndTime: Cardinal;	Время до завершения торгов в секундах

TL2List = class;

TL2List = class;	Все листы объектов в боте наследуются от этого класса, списки: бафов, нпц, чаров, дропа, предметов.
ByID(ID: Cardinal; var Obj): Boolean;	Поиск объекта в списке по ID. Если объект найден, он помещается в переменную Obj.
ByName(const Name: string; var Obj): Boolean;	Поиск объекта в списке по имени. Если объект найден, он помещается в переменную Obj.
Count: integer;	Количество объектов в списке
Items(Index: integer): TL2Object;	Позволяет обратиться к объекту в списке по индексу

TL2Auction = class(TL2List)

TL2Auction = class(TL2List)	**Класс для работы с аукционом. Список заполняется в результате работы ф-ий Search и GetMySales.** **Объект класса В скриптах доступен по имени Auction.**
`function Search(const Name: string; Grade: Integer = -1; PageID: Integer = 0): Boolean;`	Grade: 0 - NG; 1 - D,.., 10 - R99.
`function SellItem(Item: TL2Item; Count, Price, Days: Cardinal; CustomName: string = ''): Boolean;`	Days: 1, 3, 5, 7
`function BuyItem(Item: TL2AucItem): Boolean;`	
`function GetMySales: Boolean;`	получить список своих лотов продажи
`function CancelItem(Item: TL2AucItem): Boolean;`	снимет предмет с продажи
`property Items: TL2AucItem;`	

TL2WareHouse = class(TL2List)

TL2WareHouse = class(TL2List)	**Лист с TL2Item внутри вархауса**
`property WHType: Word`	Тип вархауса
`property Adena: Int64`	
`Items(Index: integer): TL2Item;`	

TSpawnList = class(TL2List);

TSpawnList = class(TL2List);	**Список объектов все что имеют координаты.**
`Items(Index: integer): TL2Spawn;`	Позволяет обратиться к объекту в списке по индексу

TNpcList = class(TL2List);

TNpcList = class(TL2List);	**Список окружающих нас NPC. В скриптах доступен по имени NpcList**
`Items(Index: integer): TL2Npc;`	Позволяет обратиться к объекту в списке по индексу

TPetList = class(TL2List);

TPetList = class(TL2List);	**Список наших петов. В скриптах доступен по имени PetList**
`Items(Index: integer): TL2Pet;`	Позволяет обратиться к объекту в списке по индексу

TCharList = class(TL2List);

TCharList = class(TL2List);	**Список окружающих нас игроков. В скриптах доступен по имени CharList**
`Items(Index: integer): TL2Char;`	Позволяет обратиться к объекту в списке по индексу

TDropList = class(TL2List);

TDropList = class(TL2List);	**Список окружающего нас дропа. В скриптах доступен по имени DropList**
`Items(Index: integer): TL2Drop;`	Позволяет обратиться к объекту в списке по индексу

TSkillList = class(TL2List);

TSkillList = class(TL2List);	**Список содержащий скилы нашего персонажа. В скриптах доступен по имени SkillList**
`Items(Index: integer): TL2Skill;`	Позволяет обратиться к объекту в списке по индексу

TBuffList = class(TL2List);

TBuffList = class(TL2List);	**Список содержащий бафы объекта.**
`Items(Index: integer): TL2Buff;`	Позволяет обратиться к объекту в списке по индексу

TItemList = class(TL2List);

TItemList = class(TL2List);	**Список инвентаря.**
`Items(Index: integer): TL2Item;`	Позволяет обратиться к объекту в списке по индексу

TParty = class;

TParty = class;	Класс описывающий нашу группу. В скриптах доступен по имени Party.
Pets: TNpcList;	Список петов в группе.
Chars: TCharList;	Список чаров в группе.
LootType: TLootType;	Тип распределения лута в группе.
Leader: TL2Char;	Лидер группы

TInventory = class;

TInventory = class;	Класс содержащий инвентари. В скриптах доступен по имени Inventory.
Pet: TItemList;	Инветарь нашего пета
User: TItemList;	Инветарь нашего персонажа.
Quest: TItemList;	Инветарь нашего персонажа (квестовый)

TConfirmDlg = class

TConfirmDlg = class	Класс подробно описывающий диалоги
MsgID : Cardinal;	
ReqID : Cardinal;	
Sender: string;	
EndTime: Cardinal;	
Valid : Boolean;	

ChatMessage

ChatMessage	Объект для работы с чатом
ChatMessage.unread: boolean;	Не прочитано нами?
ChatMessage.sender: string;	Кто отправил.
ChatMessage.text: string;	Текст что напечатал отправивший
ChatMessage.chattype: TMessageType;	Тип отправленного сообщения.

TL2Script = class

TL2Script = class;	Класс доступный по имени Script
MainProc(Proc: Pointer);	Вызов процедуры в основном потоке программы. Proc - указатль на процедуру потока; Parameter - любой передаваемый параметр (не обязательный)
NewThread(Proc: TThreadFunc; Parameter: Pointer = nil);	Создает новый поток скрипта. Proc - указатль на процедуру потока (procedure MyThread(Prm: Pointer)) Parameter - любой передаваемый параметр (не обязательный)
Path: String;	Полный путь к файлу скрипта.
PaxFile : String;	Путь скрипта.
Replace(const FullPath : String);	Заменить скрипт
Resume;	Возобновляет работу скрипа
Suspend;	Ставит скрипт на паузу

Глобальные функции и переменные:

Функции API бота:

Глобальные функции	Доступны в любом месте
`PlaySound(const FileName: string; Loop: Boolean);`	Проигрывает звуковой файл в формате wave (.wav). Loop - зациклить воспроизведение.
`StopSound;`	Останавливает воспроизведение звукового файла
`Delay(ms: Cardinal): Boolean;`	Задержка скрипта на указанное число миллисекунд.
`ExePath: string;`	Возвращает путь к папке с Adrenaline
`GetHWID : Cardinal;`	Получить уникальный код компьютера.
`GetControl(const Name: string): TL2Control;`	Получает объект Engine другого персонажа. Name - имя персонажа.
`BotLoginID: Cardinal;`	Получает уникальный ID (хэш) основанный на бот-логине.

Глобальные переменные

Глобальные переменные	доступные без импорта модулей и из любього места
`Engine : TL2Control;`	Контроллер текущего окна
`User : TL2User;`	Персонаж для текущего Engine
`Auction : TL2Auction;`	Лист с предметами на аукционе
`SpawnList : TSpawnList;`	
`NpcList : TNpcList;`	
`PetList : TPetList;`	
`CharList : TChatList;`	
`DropList : TDropList;`	
`SkillList : TSkillList;`	
`ItemList : TItemList;`	
`Party : TParty;`	
`Inventory : TInventory;`	**содержащий инвентари**
`Script : Tscript;`	
`WareHouse: TL2WareHouse;`	Лист с итемами вархуса
`ChatMessage : TChatMessage;`	буфер где хранится последняя информация о чате

Функции преобразования данных

Функции преобразования данных
`MemToHex(const dt; size: Word; sep: char = #0): String; overload;`
`MemToHex(const Mem: AnsiString): String; overload;`
`HexToMem(const Hex: string; var Buf): Cardinal; overload;`
`HexToMem(const Hex: string): AnsiString; overload;`

API для каптчи

Captcha API	
`function GetCaptcha(ImageData: TMemoryStream; APIKey: String; var CaptchaRes: String; MinLen: integer=0; MaxLen: integer=0; Numeric: integer=0; Phrase: integer=0; RegSense: integer=0; Calc: integer=0; Russian: integer=0): Integer; overload;`	Функция GetCaptcha - распознает картинку используя сервис AntiGate.com Параметры функции: ImageData - стрим, содержащий картинку с каптчей ImageFile - путь к файлу каптчи APIKey - ключ сервиса AntiGate.com для распознавания CaptchaRes - буффер, в который попадает текст каптчи, либо сообщение об ошибке MinLen - минимальная длина текста каптчи MaxLen - максимальная длина каптчи Phrase - если 1, помечает что каптча состоит из нескольких слов Regsense - если 1, помечает что текст каптчи чувствителен к регистру Numeric - если 1, помечает что текст каптчи состоит только из цифр, 2 помечает что на каптче нет цифр Calc - если 1, помечает что цифры на каптче должны быть высчитаны Russian - если 1, помечает что вводить нужно только русский текст, 2 - русский или английский Функция возвращает: Номер каптчи CaptchaID, при ошибке 0
`function GetCaptcha(ImageFile: string; APIKey: String; var CaptchaRes: String; MinLen: integer=0; MaxLen: integer=0; Numeric: integer=0; Phrase: integer=0; RegSense: integer=0; Calc: integer=0; Russian: integer=0): Integer; overload;`	
`function CaptchaBalance(APIKey: String): String;`	Функция CapchaBalance - выводит текущий баланс Параметры функции: APIKey - ключ сервиса AntiGate.com для распознавания Функция возвращает: Строку с содержанием баланса, либо 'N/A' при ошибке
`function GetCaptchaReportBad(APIKey: String; CaptchaID: Integer): String;`	Функция CapchaReportBad - отправляет жалобу о неверно распознанной каптче Параметры функции: APIKey - ключ сервиса AntiGate.com для распознавания CaptchaID - номер каптчи, которая была неверно распознана Функция возвращает: Строку с результатом ответа от сервиса (OK_REPORT_RECORDED - в случае успеха), либо 'N/A' при ошибке
`CaptchaServer`	по умолчанию равен AntiGate.com, при необходимости его можно изменить

Прочее

Прочее	**Для опытных скриптеров.**
`Procedure OnFree;`	Вызывается при завершении скрипта. Тело процедуры надо писать самому.
`function OnEntry(var Param)`	Функция, тело которой должно быть написанно в вашем коде. Для вызова с другого TL2Control, через Entry
`ShMem: array[0..1000] of integer;`	Массив, который доступен из любого TL2Control (любого персонажа)

Перечисляемые типы.

Enum values	
TL2Status = (	
lsOff,	Отключен
lsOnline,	Онлайн
lsOffline	Оффлайн
);	

TL2Race = (
rtHuman,
rtElf,
rtDarkelf,
rtOrc,
rtDwarf,
rtKamael
);

TLootType = (	
ldLooter	Нашедшому
ldRandom	Случайно
ldRandomSpoil	Случайно + присвоить
ldOrder	По очереди
ldOrderSpoil	По очереди + присвоить
);	

TStoreType = (	Тип торговой лавки
stNone,	
stSell,	
stPrepare,	
stBuy,	
stUnknown2,	
stManufacture,	
stUnknown6,	
stObservingGames,	
stSellPackage	
);	

TL2Actions = (	Перечисляемый тип для WaitAction
laSpawn,	отреспился предмет, любой TL2Spwan
laDelete,	
laPetSpawn,	
laPetDelete,	
laInvite,	
laDie,	
laRevive,	
laTarget,	Любой TL2Live взял кого то в таргет
laUnTarget,	
laInGame	в игре
laStatus	поменялся статус
laBuffs,	
laSkills,	
laDlg,	
laConfirmDlg,	p1 : TConfirmDlg
laStop,	
laStartAttack,	
laStopAttack,	
laCast,	Любой TL2Live кастует скил.
laCancelCast,	
laTeleport,	TL2Live телепортировался
laAutoSoulShot,	
laNpcTrade,	
laSysMsg,	p1 = id системного сообщения
laChat,	не юзать, вместо него ChatMessage
laKey	p1 = id клавиши
);	

```
TRestartType = (                                    для функции TL2Control.GoHome
 rtTown,
 rtClanHoll,
 rtCastle,
 rtFort,
 rtFlags
);
```

```
TMessageType = (                                    для ChatMessage.Type
 mtSystem,
 mtAll,
 mtPrivate,
 mtParty,
 mtClan,
 mtFriend,
 mtShout
);
```

```
TL2Class = (                                        для поля у TL2Object
 lcError,
 lcDrop,
 lcNpc,
 lcPet,
 lcChar,
 lcUser,
 lcBuff,
 lcSkill,
 lcItem
);
```

Классы в подключаемых модулях.

TTCPBlockSocket = class

Класс доступен в модуле TCP.

TTCPBlockSocket = class	Класс для работы с TCP соеденением
`constructor Create;`	
`destructor Destroy; override;`	
`procedure CloseSocket; override;`	
`function WaitingData: integer; override;`	
`procedure Listen; override;`	
`function Accept: integer; override;`	
`procedure Connect(IP, Port: string); override;`	
`function SendBuffer(Buffer : pointer; Len : integer) : integer; override;`	
`Function RecvBuffer(Buffer : pointer; Len : integer; Timeout : integer) : integer; virtual;`	
`Procedure SendByte(Data : byte); virtual;`	
`Function RecvByte(Timeout : integer) : byte; virtual;`	
`Procedure SendString(Data : ansistring); virtual;`	
`Function RecvString(Timeout : integer) : ansistring; virtual;`	
`Procedure SendInteger(Data : integer); virtual;`	
`Function RecvInteger(Timeout : integer) : integer;`	
`Property LastError : integer;`	

Класс доступен в модуле ICQ.

TICQ = class	Класс для работы с ICQ.
`constructor Create;`	
`destructor Destroy; override;`	
`property Status: Cardinal;`	Текущий статус: ONLINE = $00000000; INVISIBLE = $00000100; AWAY = $00000001; NA = $00000005; OFFLINE = $FFFFFFFF;
`function Connected: Boolean;`	Подключены или нет в текущий момент к серверу ICQ.
`function Connect(UIN : Cardinal; const Password : string; const Server : string = 'login.icq.com'; Port : Word = 5190; TimeOut : Byte = 5) : Boolean;`	Осуществляет подключение к серверу ICQ. UIN - номер ICQ под которым требуется войти. Password - пароль от ICQ. Server - адрес сервера ICQ. Port - порт сервера. TimeOut - время на попытку подключения (в секундах).
`procedure Disconnect;`	Отключение от сервера.
`procedure SendMessage(UIN : Cardinal; const Msg : string);`	Отправка сообщения. UIN - номер получателя. Msg - текст сообщения.
`procedure OnError(Sender : TObject; ErrorType : TErrorType; const ErrorMsg : string); virtual;`	Вызывается при ошибке. ErrorType - тип ошибки ErrorMsg - текст ошбки (Используется в наследующих классах для перекрытия - override)
`procedure OnMessageRecv(Sender : TObject; Msg, UIN : string); virtual;`	Получено сообщение. Msg - текст сообщения. UIN - номер отправителя. (Используется в наследующих классах для перекрытия - override)
`procedure OnUserOffline(Sender : TObject; UIN : string); virtual;`	Контакт из вашего списка ICQ отключился от сервера. UIN - номер отключившегося (Используется в наследующих классах для перекрытия - override)
`procedure OnServerDisconnect(Sender : TObject; Reason : LongInt; Description : string); virtual;`	Вы были отключены сервером. Reason - причина (код) Description - текстовое описание причины. (Используется в наследующих классах для перекрытия - override)

ПРИМЕРЫ

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.
```

Печатать в системном окне бота информацию о сообщении, которую написал игрок, находящийся в зоне и не ваш сопартиец.

Function TL2Live.AbnormalID

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

//Цель в УД стоит?
begin
      if isUD(User.target) then
            print('Да в УД');
end.
```

Проверить находится цель в УД или нет. Значение `$8000000` это один из битов переменной поля AbnormalID, обозначающий есть ли эффект UD.

Некоторые другие эффекты:

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

function TL2Control.GetSkillList

Для текущего скрипта **TL2Control** доступен по имени Engine так же список скилов доступен по переменной **SkillList**. С другого **TL2Control** можно получить список скиллов с помощью **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('MoyaEEshka',1255)) then
            GetControl('MoyaEEshka').UseSkill(1255) //Recall
      Else
            Engine.UseItem(736); //Юзаем сое
end.
```

Если есть у нашей ЕЕ с ником `'MoyaEEshka'` скилл Party Recall, то используем его на ней, если нету то используем сое.

function TL2Control. GameWindow

Зачем он нужен этот HWND игрового окна?
Например в скрипте можно использовать WinApi функции.

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

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

развернуть окно с игрой на передний план, если мы померли.

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

Чтобы раскрасить системный чат в Боте можно использовать данный метод.
Цвет задается целочисленным значением. Некоторые из возможных:

```
clBlack = 0;  //черный
clMaroon = 128; //Тёмно-красный
clGreen = 32768; //Зелёный
clOlive = 32896; //Оливковый
clNavy = 8388608; //Тёмно-синий
clPurple = 8388736; //Пурпурный
clTeal = 8421376; //Стальной
clGray = 8421504; //Серый
clSilver = 12632256; //Серебряный
clRed = 255; //Красный
clLime = 65280; //Ярко-зелёный
clYellow = 65535; //Жёлтый
clBlue = 16711680; //Синий
clFuchsia = 16711935; //Фиолетовый
clAqua = 16776960; //Бирюзовый
clWhite = 16777215; //Белый
```

TL2Live.Teleport...

Знать информацию о каждом объекте, когда телепортировался и насколько далеко. В том числе и о себе.

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

Проиграть звук при телепорте из папки с ботом, в папке Sounds звук dc.wav.

Отключить интерфейс на 15 сек всех кто в пати и на одном боте, при телепорте

```
procedure FaceControlForAll(Pos : Boolean);
    var
        NowEng : TL2Control;
        i : Integer;
    begin
        For i := 0 to Party.Chars.Count - 1 do
                begin
                    NowEng := GetControl(Party.Chars.items(i).name);
                    if Assigned(NowEng) then
                        NowEng.FaceControl(0, Pos);
                end;
        Engine.FaceControl(0, Pos);
    end;

procedure MassOff;
    var
        P1, P2 : Pointer;
    begin
        While True do
          begin
            if (Engine.WaitAction([laSysMsg], P1, P2, 500) = laTeleport) and
               (TL2Char(P1).IsMember or
               (TL2Char(P1) = User)) then
            begin
                FaceControlForAll(False);
                Delay(15000);
                FaceControlForAll(True);
            end;
          end;
    end;

//**************************

begin
  Script.NewThread(@MassOff);
  Delay (-1);
end.
```

в методе FaceControlForAll пробегаемся циклом по всем никами игроков в пати, где случился телепорт, в случае если игрок запущен на этом компьютере (боте) выключаем его.

Простейший делевел

```
var
        I : INteger;
        dist : Cardinal;
        TrueGvard, CheckGvard : TL2Npc;
const
        GvardsArray : array [1..8] of Cardinal =
            (30039, 30040, 30041, 30042, 30043, 30044, 30045, 30046);
begin
        print('Начинаем Делевел');

        While User.level > 81 do
            begin
                dist := 10000;
                TrueGvard := nil;
                For i := low(GvardsArray) to High(GvardsArray) do
                    if NpcList.ByID(GvardsArray[i], CheckGvard) and (User.Distto(CheckGvard) < Dist) then
                        begin
                            TrueGvard := CheckGvard;
                            dist := User.Distto(CheckGvard);
                        end;
                if Assigned(TrueGvard) then
                    begin
                        Engine.MoveTo(TrueGvard, -5);
                        Engine.SetTarget(TrueGvard);
                        Engine.Attack(500, True);
                    end;
                if User.dead then
                    Engine.GoHome;
            end;
end.
```

Если видим айди нпц который указан в массиве, GvardsArray (по умолчанию вписанны гварды с острова Людей),
находим самого ближнего.
бежим к нему в плотную, и начинаем через контрл атаковать.
Если померли встаем в город.

Автоматически кидаем репорт игроку взявшему нас в таргет

```
uses
    SysUtils, Classes;

var
    TargetLogSL : TStringList;

Procedure CheckTarget;
    var
        i : Integer;
    begin
        if TargetLogSL.Count > 5 then exit;

        i := 0;
            While i < CharList.Count do
                begin
                    if ((CharList.Items(i).target.OID = User.OID) or CharList.Items(i).target.IsMember) and not
CharList.Items(i).IsMember then
                        if TargetLogSL.IndexOf(CharList.Items(i).name) = -1 then
                            begin
                                TargetLogSL.Add(CharList.Items(i).name);
                                print('нас затаргетил: ' + CharList.Items(i).name);
                                if Engine.SetTarget(CharList.Items(i)) then
                                    Engine.UseAction(65);
                                break;
                            end;
                    inc(i);
                end;
    end;

Procedure OnFree;
    begin
          TargetLogSL.Free;
    end;

begin
    TargetLogSL := TStringList.Create;
    While Engine.Delay(100) do
        CheckTarget;
    TargetLogSL.Free;

end.
```

Перехват нажатия клавиш, вместе с ctrl alt shift.

```
1.    uses
2.        SysUtils;
3.
4.    function GetKeyState(nVirtKey: integer): byte; stdcall;
5.          external 'user32.dll' name 'GetKeyState' ;
6.
7.    function KeyDown(K : byte) : Boolean;
8.        begin
9.            Result := (K = 128) or (K = 129)
10.       end;
11.
12.   function SHIFT : boolean;
13.       begin
14.           Result := KeyDown(GetKeyState($10));
15.       end;
16.
17.   function CTRL : boolean;
18.       begin
19.           Result := KeyDown(GetKeyState($11));
20.       end;
21.
22.   function ALT : boolean;
23.       begin
24.           Result := KeyDown(GetKeyState($12));
25.       end;
26.
27.   procedure ActionThread(p: pointer); //Поток для ожидания/обработки событий
28.   var
29.       KeyCode, i: integer;
30.   begin
31.       while Engine.Status = lsOnline do
32.           begin  //Цикл действует пока чар в игре (Online)
33.               Engine.WaitAction([laKey], KeyCode, i);
34.
35.               if ctrl and shift and (KeyCode = $4F) then //ctrl + SHift + "O"
36.                       Engine.UseSkill(1398);
37.
```

```
38.                 if ctrl and alt and (KeyCode = $50) then //ctrl + alt + "P"
39.                         Engine.UseSkill(1398);
40.             end;
41.     end;
42.
43.     begin
44.       Script.NewThread(@ActionThread);
45.       Delay(-1); //Бесконечная пауза
46.     end.
```