

Реестр

- **HKEY_CLASSES_ROOT**: Эта ветвь содержит все типы Ваших ассоциаций к файлам, информацию об OLE и данные по ярлыкам.
- **HKEY_CURRENT_USER**: Эта ветвь связана с ветвью HKEY_USERS, и соответствует пользователю, работающему в настоящее время на PC.
- **HKEY_LOCAL_MACHINE**: Эта ветвь содержит определенную информацию о типах аппаратных средств, программного обеспечения, и других настройках на данном PC, эта информация используется для всех пользователей, которые работают на этом компьютере.

- **HKEY_USERS**: Эта ветвь содержит индивидуальные настройки каждого пользователя компьютера, каждый пользователь представлен под ключом SID, расположенном под главной ветвью.
- **HKEY_CURRENT_CONFIG**: Эта ветвь связана с ветвью HKEY_LOCAL_MACHINE, и соответствует текущей аппаратной конфигурации.
- **HKEY_DYN_DATA**: Эта ветвь связана с частью HKEY_LOCAL_MACHINE, и служит для использования особенностей Plug-&Play в Windows, этот раздел динамически изменяется, когда устройства добавляются и удаляются из системы.

функции API

- **RegCreateKey (Key: HKey; SubKey: PChar; var Result: HKey): Longint;**
- Создать подраздел в реестре. Key указывает на "корневой" раздел реестра. SubKey - имя раздела - строится по принципу пути к файлу в DOS (пример subkey1\subkey2\ ...). Если такой раздел уже существует, то он открывается (в любом случае при успешном вызове Result содержит Handle на раздел). Об успешности вызова судят по возвращаемому значению, если ERROR_SUCCESS, то успешно, если иное - ошибка.

- **RegOpenKey(Key: HKey; SubKey: PChar; var Result: HKey): Longint;**
- Открыть подраздел Key\SubKey и возвращает Handle на него в переменной Result. Если раздела с таким именем нет, то он не создается. Возврат - код ошибки или ERROR_SUCCESS, если успешно.

- **RegCloseKey(Key:HKey):Longint;**
- Закрывает раздел, на который ссылается Key. Возврат - код ошибки или ERROR_SUCCESS, если успешно.

- **RegDeleteKey(Key: HKey; SubKey: PChar): Longint;**
- Удалить подраздел Key\SubKey. Возврат - код ошибки или ERROR_SUCCESS, если нет ошибок.

- **RegEnumKey(Key: HKey; index: Longint; Buffer: PChar;cb: Longint): Longint;**
- Получить имена всех подразделов раздела Key, где Key - Handle на открытый или созданный раздел (см. RegCreateKey и RegOpenKey), Buffer - указатель на буфер, cb - размер буфера, index - индекс, должен быть равен 0 при первом вызове RegEnumKey. Типичное использование - в цикле While, где index увеличивается до тех пор, пока очередной вызов RegEnumKey не завершится ошибкой

- **RegQueryValue(Key: HKey; SubKey: PChar; Value: PChar; var cb: Longint): Longint;**
- Возвращает текстовую строку, связанную с ключом Key\SubKey.Value - буфер для строки; cb- размер, на входе - размер буфера, на выходе - длина возвращаемой строки. Возврат - код ошибки.

- **RegSetValue(Key: HKey; SubKey: PChar; ValType: Longint; Value: PChar; cb: Longint): Longint;**
- Задать новое значение ключу Key\SubKey, ValType - тип задаваемой переменной, Value - буфер для переменной, cb - размер буфера. В Windows 3.1 допустимо только Value=REG_SZ. Возврат - код ошибки или ERROR_SUCCESS, если нет ошибок.

TRegIniFile

```
procedure TForm1.Button1Click(Sender: TObject);  
var RegIni:TRegIniFile;  
Begin  
{ создаётся переменная RegIni типа  
  TRegIniFile в скобках указываем имя  
  подраздела, с которым собираемся  
  работать}  
RegIni:=TRegIniFile.Create('Software');  
RegIni.OpenKey('online', true);  
RegIni.WriteString('Razd', 'Param', Edit1.Text);  
RegIni.Free;end;
```

- Первая строка инициализирует реестр и текущем разделе становится `HKEY_CURRENT_USER\Software`. Если бы мы написали нашу первую строку так: `RegIni:=TRegIniFile.Create(' Network')` то текущим разделом был бы уже: `HKEY_CURRENT_USER\Network`. Затем открываем ещё подраздел: `RegIni.OpenKey('online', true)`. `OpenKey` - открывает следующий подраздел, имя которого указано в качестве первого параметра в скобках. Второй параметр означает: создавать ли раздел если его нет? Если `true` - это значит, что когда такого подраздела нет, то он будет создан

- Функция `WriteString` записывает строку в текущий раздел реестра. Первый параметр - имя подраздела в текущем разделе, т.е. наше значение будет записываться в `HKEY_CURRENT_USER\Software\VR-online\Razd`.
- Вторым параметром - имя записываемого параметра, а третий - значение параметра. В качестве значения мы используем строку записанную в `Edit1`.
- Последняя строка `RegIni.Free` уничтожает созданную нами переменную.
- Для записи в реестр мы использовали процедуру **`WriteString`**. Но она не единственная, есть ещё **`WriteBool`** (для записи булево значения) и **`WriteInteger`** (для записи целого значения).

Чтение

```
procedure TForm1.Button2Click(Sender: TObject);  
var RegIni:TRegIniFile;  
begin RegIni:=TRegIniFile.Create('Software');  
    RegIni.OpenKey('online', true);  
    Edit1.Text:=RegIni.ReadString('Razd','Param','Default');  
RegIni.Free;  
end;
```

RootKey

это свойство объекта TRegIniFile, которое указывает на головной раздел, который сейчас используется. По умолчанию это HKEY_CURRENT_USER. Чтобы изменить это значение, нужно просто присвоить другое.

```
RegIni.RootKey:=HKEY_LOCAL_MACHINE;
```