

GrdDeCode

Функция(метод) **GrdDeCode** декодирует блок данных быстрым взаимнообратным преобразованием.

Функция(или метод) **GrdDeCode** были предназначены для работы с устаревшими ключами **Guardant Stealth**. Данная функция(или метод) реализованы исключительно в целях совместимости, и использование их в современных приложениях не рекомендуется. Вместо **GrdDeCode** рекомендуется использовать функцию(метод) [GrdCrypt](#).

C

```
int GRD_API GrdDeCode(
    DWORD dwCnvType,
    void *pKeyBuf,
    void *pData,
    DWORD dwLng
);
```

dwCnvType	метод быстрого взаимнообратного преобразования. Задается одним из флагов GrdAT_XXX. Должен быть эквивалентен методу, заданному функцией GrdCodeInit									
	GrdAT_Algo0	Базовый метод. Кодирование выполняется блоками по 32 байта. Этот метод лучше всего использовать для преобразования строковых данных. Важная особенность метода: если закодировать большой объем памяти (или файл) участками, то декодирование нужно будет обязательно делать точно теми же участками, или же размер участков должен быть кратен 32 байтам. Функции GrdEnCode и GrdDeCode не изменяют пароль.								
	GrdAT_AlgoASCII	Символьный метод. Кодирование выполняется блоками по 32 байта. Этот метод хорош для кодирования строковых данных. Например, если вы хотите закодировать название поля базы данных, указанное в исходном тексте программы, то в некоторых языках программирования будет проблематично создать строки с несимвольными значениями. Символьный метод кодирует строки так, чтобы в них не было неотображаемых символов. Функции GrdEnCode и GrdDeCode не изменяют пароль.								
	GrdAT_AlgoFile	Файловый метод. Кодирование выполняется блоками произвольной длины, поэтому такой метод подходит для кодирования файлов. Кодирование этим методом файла блоками меньше 32 байт неэффективно. Функции GrdEnCode и GrdDeCode в этом методе преобразования изменяют пароль, поэтому последовательность использования функции GrdDeCode должна совпадать с последовательностью использования функции GrdEnCode.								
pKeyBuf	буфер, содержащий преобразованный пароль									
pData	буфер с данными для раскодирования									
dwLng	объем (в байтах) данных для раскодирования									
	<table><tr><td>dwCnvType</td><td>метод быстрого взаимнообратного преобразования (должен быть эквивалентен методу, заданному функцией GrdCodeInit)</td></tr><tr><td>pKeyBuf</td><td>Адрес переменной, содержащей преобразованный пароль</td></tr><tr><td>pData</td><td>Адрес буфера с данными для декодирования</td></tr><tr><td>dwLng</td><td>Объем (в байтах) данных для декодирования</td></tr></table>		dwCnvType	метод быстрого взаимнообратного преобразования (должен быть эквивалентен методу, заданному функцией GrdCodeInit)	pKeyBuf	Адрес переменной, содержащей преобразованный пароль	pData	Адрес буфера с данными для декодирования	dwLng	Объем (в байтах) данных для декодирования
dwCnvType	метод быстрого взаимнообратного преобразования (должен быть эквивалентен методу, заданному функцией GrdCodeInit)									
pKeyBuf	Адрес переменной, содержащей преобразованный пароль									
pData	Адрес буфера с данными для декодирования									
dwLng	Объем (в байтах) данных для декодирования									
dwCnvType	Метод быстрого взаимнообратного преобразования. Задается одним из флагов GrdAT_XXX									
GrdAT_Algo	GrdAT_Algo0	Базовый метод. Кодирование выполняется блоками по 32 байта. Этот метод лучше всего использовать для преобразования строковых данных. Важная особенность метода: если закодировать большой объем памяти (или файл) участками, то декодирование нужно будет обязательно делать точно теми же участками, или же размер участков должен быть кратен 32 байтам. Функции GrdEnCode и GrdDeCode не изменяют пароль.								
	GrdAT_AlgoASCII	Символьный метод. Кодирование выполняется блоками по 32 байта. Этот метод хорош для кодирования строковых данных. Например, если вы хотите закодировать название поля базы данных, указанное в исходном тексте программы, то в некоторых языках программирования будет проблематично создать строки с несимвольными значениями. Символьный метод кодирует строки так, чтобы в них не было неотображаемых символов. Функции GrdEnCode и GrdDeCode не изменяют пароль.								
	GrdAT_AlgoFile	Файловый метод. Кодирование выполняется блоками произвольной длины, поэтому такой метод подходит для кодирования файлов. Кодирование этим методом файла блоками меньше 32 байт неэффективно. Функции GrdEnCode и GrdDeCode в этом методе преобразования изменяют пароль, поэтому последовательность использования функции GrdDeCode должна совпадать с последовательностью использования функции GrdEnCode.								

Возможные ошибки

GrdE_InvalidCnvType	Указан неверный метод преобразования
	Набор ошибок Guardant API

Функция **GrdDeCode** позволяет раскодировать информацию, ранее закодированную функцией [GrdEnCode](#), используя быстрое взаимнообратное преобразование. При этом функция не обращается к ключу - преобразование полностью производится компьютером. За счет этого и достигается высокая скорость преобразования.

Параметр *pKeyBuf* содержит пароль, преобразованный ранее функцией [GrdCodeInit](#). С его помощью и будет производиться раскодирование данных. Сами данные должны быть расположены по адресу, заданному параметром *pData*, длина раскодировемой последовательности (в байтах) задается параметром *dwLng*. Раскодирование производится по методу, указанному в параметре *wCnvType*. Выбирая тот или иной метод преобразования необходимо учитывать его особенности и ограничения. Метод кодирования должен соответствовать заданному при вызове функции [GrdCodeInit](#).

При успешном завершении по адресу, указанному в *pData*, будут помещены раскодированные данные. В этом случае функция вернет [GrdE_OK](#).

C#

```
public static GrdE GrdDeCode(GrdAT cnvType, byte[] key, byte[] data)
```

cnvType [in]

Тип: [GrdAT](#)

Метод быстрого взаимнообратного преобразования. Задается одним из флагов [GrdAT](#).

key [in]

Тип: byte []

Буфер, который содержит преобразованный пароль.

data [in]

Тип: byte []

Буфер с данными для раскодирования.

Возможные ошибки

GrdE.InvalidCnvType	Указан неверный метод преобразования
	Набор ошибок Guardant API

Метод **GrdDeCode** позволяет раскодировать информацию, ранее закодированную функцией [GrdEnCode](#), используя быстрое взаимнообратное преобразование. При этом метод не обращается к ключу - преобразование полностью производится компьютером. За счет этого и достигается высокая скорость преобразования.

Параметр *key* содержит пароль, преобразованный ранее методом [GrdCodeInit](#). С его помощью и будет производиться раскодирование данных. Сами данные должны быть расположены по адресу, заданному параметром *data*. Раскодирование производится по методу, указанному в параметре *cnvType*. Выбирая тот или иной метод преобразования необходимо учитывать его особенности и ограничения. Метод кодирования должен соответствовать заданному при вызове функции [GrdCodeInit](#).

При успешном завершении по адресу, указанному в *data*, будут помещены раскодированные данные. В этом случае метод вернет [GrdE.OK](#).

Java

```
public static GrdE GrdDeCode(int cnvType, byte[] key, byte[] data)
```

cnvType [in]

Тип: int

Метод быстрого взаимнообратного преобразования.

key [in]

Тип: byte []

Буфер, который содержит преобразованный пароль.

data [in]

Тип: byte []

Буфер с данными для раскодирования.

Возможные ошибки

GrdE.InvalidCnvType	Указан неверный метод преобразования
	Набор ошибок Guardant API

Метод **GrdDeCode** позволяет раскодировать информацию, ранее закодированную функцией [GrdEnCode](#), используя быстрое взаимнообратное преобразование. При этом метод не обращается к ключу - преобразование полностью производится компьютером. За счет этого и достигается высокая скорость преобразования.

Параметр *key* содержит пароль, преобразованный ранее методом [GrdCodeInit](#). С его помощью и будет производиться раскодирование данных. Сами данные должны быть расположены по адресу, заданному параметром *data*. Раскодирование производится по методу, указанному в параметре *slvType*. Выбирая тот или иной метод преобразования необходимо учитывать его особенности и ограничения. Метод кодирования должен соответствовать заданному при вызове функции [GrdCodeInit](#).

При успешном завершении по адресу, указанному в *data*, будут помещены раскодированные данные. В этом случае метод вернет [GrdE.OK](#).