

GrdPI_GetTimeLimit

Функция(метод) **GrdPI_GetTimeLimit** возвращает оставшееся время работы алгоритма или защищенной ячейки.

C

```
int GrdPI_GetTimeLimit(
    HANDLE hGrd,
    DWORD dwItemNum,
    TGrdSystemTime *pGrdSystemTime,
    void *pReserved
);
```

<i>hGrd</i>	хэндл, через который будет выполнена данная операция
<i>dwItemNum</i>	числовое имя аппаратного алгоритма/защищенной ячейки
<i>pGrdSystemTime</i>	указатель на структуру TGrdSystemTime : <pre>typedef struct { WORD wYear; // The year (2000 - 2099) WORD wMonth; // The month (January = 1, February = 2, ...) WORD wDayOfWeek; // The day of the week (Sunday = 0, Monday = 1, ...) WORD wDay; // The day of the month (1-31) WORD wHour; // The hour (0-23) WORD wMinute; // The minute (0-59) WORD wSecond; // The second (0-59) WORD wMilliseconds; // The millisecond (0-999) } TGrdSystemTime;</pre>
<i>pReserved</i>	зарезервировано, должно быть равно NULL

GrdE_OK	нет ошибок
GrdE_NeedInitialization	требуется инициализация API (вызов GrdStartup)
GrdE_InvalidHandle	недействительный хэндл
GrdE_NoService	для алгоритма/ячейки сервис не предусмотрен
GrdE_InvalidArg	недопустимый параметр при вызове функции

Функция **GrdPI_GetTimeLimit** возвращает оставшееся время работы алгоритма или защищенной ячейки. Числовое имя алгоритма или защищенной ячейки задается параметром *dwItemNum*. Для работы требуется предварительно установить **Private Read Code** при помощи функции **GrdSetAccessCodes**. Если время работы алгоритма неограниченно, возвращается ошибка **GrdE_NoService**.

При использовании механизма **DeadTime** (а также и для **LifeTime**, т. к. после активации алгоритма логика работы механизма времени полностью идентична **DeadTime**) с Time-ключами, оставшееся число дней до активации будет отображаться для обратного порядка месяцев (при этом общее время до деактивации будет рассчитано корректно).

К примеру:

При задании

- **времени ключа:** 1 января 2008, 00-00-00,
 - **времени деактивации ключа:** 1 марта 2008, 00-00-03 (либо **время жизни** 2 месяца 3 секунды):
1. Если используется **DeadTime**-механизм, то после первого вызова **GrdPI_GetTimeLimit** вернется 2 месяца 2 секунды, после вызова **GrdPI_GetTimeLimit** через 3 секунды вернется 1 месяц, 28 дней, 23 часа, 59 минут, 59 секунд. Т. о. произошел "размен" февраля, а не текущего месяца.
 2. Если используется **LifeTime**-механизм, то в зависимости поля state структуры **LifeTime**:
 - **1** (алгоритм активирован): все как в п.1.
 - **0** (алгоритм не активирован): до первого вызова **GrdTransform** будет возвращаться 2 месяца, 3 секунды, после него - как в п.1.

Примечательно, что пересчет структуры времени происходит в момент активации алгоритма (при первом вызове **GrdTransform**). К примеру, при использовании механизма **LifeTime**, состояния "0"(не активирован), результатом задания времени жизни, к примеру 255 секунд, до активации алгоритма функция **GrdPI_GetTimeLimit** будет возвращать 255 секунд, после активации - произойдет пересчет в минуты и секунды.

Т. о. при проектировании ПО рекомендуется опираться на совокупность значений оставшегося до деактивации времени, нежели на отдельное число дней.

При указании нулевого времени жизни алгоритма (механизм **LifeTime**) после первого вызова **GrdTransform** алгоритм деактивируется не моментально, а через ~секунду.

C#

```
public static GrdE GrdPI_GetTimeLimit(Handle grdHandle, GrdAlgNum algNum, out GrdSystemTime systemTime)
```

grdHandle [in]

Тип: [Handle](#)

Хэндл, через который будет выполнена данная операция.

algNum [in]

Тип: [GrdAlgNum](#)

Числовое имя аппаратного алгоритма/защищенной ячейки.

systemTime [out]

Тип: [GrdSystemTime](#)

Указатель на структуру GrdSystemTime

GrdE.OK	нет ошибок
GrdE.NeedInitialization	требуется инициализация API (вызов GrdStartup)
GrdE.InvalidHandle	недействительный хэндл
GrdE.NoService	для алгоритма/ячейки сервис не предусмотрен
GrdE.InvalidArg	недопустимый параметр при вызове функции

Метод **GrdPI_GetTimeLimit** возвращает оставшееся время работы алгоритма или защищенной ячейки. Числовое имя алгоритма или защищенной ячейки задается параметром *algNum*. Для работы требуется предварительно установить **Private Read Code** при помощи метода [GrdSetAccessCodes](#). Если время работы алгоритма неограниченно, возвращается ошибка [GrdE.NoService](#).

При использовании механизма **DeadTime** (а также и для **LifeTime**, т. к. после активации алгоритма логика работы механизма времени полностью идентична **DeadTime**) с Time-ключами, оставшееся число дней до активации будет отображаться для обратного порядка месяцев (при этом общее время до деактивации будет рассчитано корректно).

К примеру:

При задании

- **времени ключа:** 1 января 2008, 00-00-00,
 - **времени деактивации ключа:** 1 марта 2008, 00-00-03 (либо **время жизни** 2 месяца 3 секунды):
1. Если используется **DeadTime**-механизм, то после первого вызова **GrdPI_GetTimeLimit** вернется 2 месяца 2 секунды, после вызова **GrdPI_GetTimeLimit** через 3 секунды вернется 1 месяц, 28 дней, 23 часа, 59 минут, 59 секунд. Т. о. произошел "размен" февраля, а не текущего месяца.
 2. Если используется **LifeTime**-механизм, то в зависимости поля state структуры **LifeTime**:
 - **1** (алгоритм активирован): все как в п.1.
 - **0** (алгоритм не активирован): до первого вызова [GrdTransform](#) будет возвращаться 2 месяца, 3 секунды, после него - как в п.1.

Примечательно, что пересчет структуры времени происходит в момент активации алгоритма (при первом вызове [GrdTransform](#)). К примеру, при использовании механизма **LifeTime**, состояния "0"(не активирован), результатом задания времени жизни, к примеру 255 секунд, до активации алгоритма метод **GrdPI_GetTimeLimit** будет возвращать 255 секунд, после активации - произойдет пересчет в минуты и секунды.

Т. о. при проектировании ПО рекомендуется опираться на совокупность значений оставшегося до деактивации времени, нежели на отдельное число дней.

При указании нулевого времени жизни алгоритма (механизм **LifeTime**) после первого вызова **GrdTransform** алгоритм деактивируется не моментально, а через ~секунду.

Java

```
public static GrdE GrdPI_GetTimeLimit(Handle grdHandle, int itemNum, GrdSystemTime systemTime)
```

grdHandle [in]

Тип: [Handle](#)

Хэндл, через который будет выполнена данная операция.

itemNum [in]

Тип: int

Числовое имя аппаратного алгоритма/защищенной ячейки.

systemTime [out]

Тип: [GrdSystemTime](#)

Указатель на структуру GrdSystemTime

GrdE.OK	нет ошибок
GrdE.NeedInitialization	требуется инициализация API (вызов GrdStartup)
GrdE.InvalidHandle	недействительный хэндл
GrdE.NoService	для алгоритма/ячейки сервис не предусмотрен
GrdE.InvalidArg	недопустимый параметр при вызове функции

Метод **GrdPI_GetTimeLimit** возвращает оставшееся время работы алгоритма или защищенной ячейки. Числовое имя алгоритма или защищенной ячейки задается параметром *itemNum*. Для работы требуется предварительно установить **Private Read Code** при помощи метода [GrdSetAccessCodes](#). Если время работы алгоритма неограниченно, возвращается ошибка [GrdE.NoService](#).

При использовании механизма **DeadTime** (а также и для **LifeTime**, т. к. после активации алгоритма логика работы механизма времени полностью идентична **DeadTime**) с Time-ключами, оставшееся число дней до активации будет отображаться для обратного порядка месяцев (при этом общее время до деактивации будет рассчитано корректно).

К примеру:

При задании

- **времени ключа:** 1 января 2008, 00-00-00,
 - **времени деактивации ключа:** 1 марта 2008, 00-00-03 (либо **время жизни** 2 месяца 3 секунды):
1. Если используется DeadTime-механизм, то после первого вызова **GrdPI_GetTimeLimit** вернется 2 месяца 2 секунды, после вызова **GrdPI_GetTimeLimit** через 3 секунды вернется 1 месяц, 28 дней, 23 часа, 59 минут, 59 секунд. Т. о. произошел "размен" февраля, а не текущего месяца.
 2. Если используется **LifeTime-механизм**, то в зависимости поля state структуры **LifeTime**:
 - **1** (алгоритм активирован): все как в п.1.
 - **0** (алгоритм не активирован): до первого вызова [GrdTransform](#) будет возвращаться 2 месяца, 3 секунды, после него - как в п.1.

Примечательно, что пересчет структуры времени происходит в момент активации алгоритма (при первом вызове [GrdTransform](#)). К примеру, при использовании механизма **LifeTime**, состояния "0"(не активирован), результатом задания времени жизни, к примеру 255 секунд, до активации алгоритма метод **GrdPI_GetTimeLimit** будет возвращать 255 секунд, после активации - произойдет пересчет в минуты и секунды.

Т. о. при проектировании ПО рекомендуется опираться на совокупность значений оставшегося до деактивации времени, нежели на отдельное число дней.

При указании нулевого времени жизни алгоритма (механизм **LifeTime**) после первого вызова [GrdTransform](#) алгоритм деактивируется не моментально, а через ~секунду.