

GrdLogin

Функция(метод) **GrdLogin** позволяет копии защищенного приложения получить лицензию для дальнейшей работы на сервере Guardant Net или на локальном ключе Guardant.

C

```
int GRD_API GrdLogin(
    HANDLE hGrd,
    DWORD dwModuleLMS,
    DWORD dwLoginFlags
);
```

<i>hGrd</i>	хэндл, через который будет выполнена данная операция						
<i>dwModuleLMS</i>	номер модуля в таблице LMS, на котором будет регистрироваться копия приложения. Если LMS не используется, параметр должен быть равен 0xFFFFFFFF. Для локальных ключей этот параметр игнорируется						
<i>dwLoginFlags</i>	флаг, устанавливающий режим лицензирования при использовании сетевых ключей. Для локальных ключей этот параметр игнорируется <table><tr><td>GrdLM_PerStation</td><td>Сетевые лицензии распределяются рабочим станциям, вне зависимости от количества запущенных на одной станции копий приложения</td></tr><tr><td>GrdLM_PerProcess</td><td>Сетевые лицензии распределяются процессам. Каждый новый регистрируемый через GrdLogin процесс получит отдельную лицензию, безотносительно того на одном или нескольких компьютерах они работают</td></tr><tr><td>GrdLM_PerHandle</td><td>Сетевые лицензии распределяются хэндлам. Каждый новый регистрируемый через GrdLogin хэндл получит отдельную лицензию, безотносительно того на одном или нескольких компьютерах они работают</td></tr></table>	GrdLM_PerStation	Сетевые лицензии распределяются рабочим станциям, вне зависимости от количества запущенных на одной станции копий приложения	GrdLM_PerProcess	Сетевые лицензии распределяются процессам. Каждый новый регистрируемый через GrdLogin процесс получит отдельную лицензию, безотносительно того на одном или нескольких компьютерах они работают	GrdLM_PerHandle	Сетевые лицензии распределяются хэндлам. Каждый новый регистрируемый через GrdLogin хэндл получит отдельную лицензию, безотносительно того на одном или нескольких компьютерах они работают
GrdLM_PerStation	Сетевые лицензии распределяются рабочим станциям, вне зависимости от количества запущенных на одной станции копий приложения						
GrdLM_PerProcess	Сетевые лицензии распределяются процессам. Каждый новый регистрируемый через GrdLogin процесс получит отдельную лицензию, безотносительно того на одном или нескольких компьютерах они работают						
GrdLM_PerHandle	Сетевые лицензии распределяются хэндлам. Каждый новый регистрируемый через GrdLogin хэндл получит отдельную лицензию, безотносительно того на одном или нескольких компьютерах они работают						

Набор ошибок Guardant API

Функция **GrdLogin** необходима для регистрации данной копии защищенного приложения на сервере Guardant Net или на локальном ключе. Регистрация осуществляется в соответствии с критериями поиска, установленными функцией [GrdSetFindMode](#).

Если критериям поиска удовлетворяют несколько обнаруженных ключей, регистрация производится на первом ответившем из них. Для того, чтобы избежать таких ситуаций, рекомендуется однозначно задавать параметры поиска ключа через [GrdSetFindMode](#). В противном случае Вам придется при помощи последовательного поиска ключей функцией [GrdFind](#) построить список ключей, а затем функцией [GrdSetFindMode](#) последовательно устанавливать в качестве критерия ID нужного ключа, выполнять к нему **GrdLogin**, проверять тот ли это ключ и если нет, то вызывать [GrdLogout](#) и так далее по списку. Это позволит избежать ошибочного выбора ключа при регистрации, но будет исполняться гораздо медленнее.

Поэтому правильный путь - делать так чтобы ключи приложений отличались в том числе и какими-либо параметрами, которые можно задать через [GrdSetFindMode](#). Например разные приложения разными номерами программ **dwProg**, разные версии приложений полями **dwVer** и так далее.

Для сетевых ключей

В соответствии с параметрами, установленными в конфигурационном файле [сетевых настроек клиента](#) функция **GrdLogin** производит поиск доступных серверов Guardant Net, на которых зарегистрированы удаленные ключи, удовлетворяющие критериям поиска. Условия поиска ключа задаются функцией [GrdSetFindMode](#). Если такой ключ найден (т. е. информация о нем доступна серверу Guardant Net), производится уменьшение на 1 значения его сетевого ресурса, после чего регистрация данной копии приложения будет считаться выполненной (будет возвращено [GrdE_OK](#)). В противном случае будет возвращен соответствующий код ошибки, и регистрация будет отменена.

Захват лицензии осуществляется в соответствии со значением параметра *dwLoginFlags*. Если значение этого параметра установлено как **GrdLM_PerStation**, то все копии приложения, запущенные на данной рабочей станции будут использовать одну лицензию. То есть при запуске второй и последующих копий счетчик лицензий уменьшаться не будет.

Если значение *dwLoginFlags* установлено как **GrdLM_PerProcess**, то лицензии будут распределяться процессам, или копиям защищенного приложения. То есть при запуске каждого нового процесса счетчик лицензий на сервере Guardant Net будет уменьшаться на 1.

dwLoginFlags установленное как **GrdLM_PerHandle** означает, что лицензия выделяется на каждый хэндл, создаваемый Guardant API (и/или автозащитой). Если при защите приложения используется несколько экземпляров библиотеки Guardant API и, соответственно несколько хэндлов (к примеру, при комбинированном использовании Guardant API и автозащиты), то лицензии будут выделены по числу хэндлов. В случае работы приложения с единственным хэндлом (т. е. защита только Guardant API или только автоматическая) распределение по хэндлам не будет отличаться от распределения по процессам.

При использовании License Management System и при наличии в ключе таблицы лицензий при регистрации кроме получения лицензии из общего счетчика захватывается лицензия из таблицы лицензий. Номер ячейки таблицы лицензий (номер модуля) должен быть указан в параметре *dwModuleLMS*. Если LMS не используется, значение параметра должно быть 0xFFFFFFFF.

Процедура регистрации будет выполнена успешно, если сервер Guardant Net располагает свободными лицензиями, а при использовании LMS - также и свободными лицензиями для конкретного модуля.

Сервер Guardant Net будет хранить нужную ему информацию о зарегистрированной копии защищенного приложения до момента снятия ее регистрации функцией [GrdLogout](#), а все функции Guardant API для установки связи с сервером будут пользоваться его сетевым адресом, полученным при регистрации. Это существенно ускорит сетевой обмен.

Для локальных ключей

При работе с локальными ключами параметры *dwModuleLMS* и *dwLoginFlags* игнорируются, поскольку локальные ключи не поддерживают распределения лицензий.

Внимание!

1. Вызов функции **GrdLogin** открывает сеанс непосредственного общения с ключом. Эту функцию необходимо вызывать перед любой другой функцией, обращающейся непосредственно к ключу. При обращении к ключу до успешного выполнения GrdLogin будет возвращена ошибка [GrdE_NeedLogin](#).
2. Нельзя вызывать **GrdLogin** несколько раз в течение одного сеанса работы с ключом Guardant (т. е. до вызова функции [GrdLogout](#), закрывающей сеанс работы с ключом). При повторной попытке вызвать **GrdLogin** будет возвращена ошибка [GrdE_NeedLogout](#).
3. Не следует выполнять **GrdLogin** и [GrdLogout](#) без необходимости, это замедляет работу защищенного приложения. Рекомендуется придерживаться следующего порядка работы: приложение стартует (начало сессии работы с приложением) - вызов **GrdLogin**, приложение завершается (конец сессии работы с приложением) - вызов [Logout](#).

C#

```
public static GrdE GrdLogin(Handle grdHandle, GrdLM loginFlags)
public static GrdE GrdLogin(Handle grdHandle, int moduleLMS, GrdLM loginFlags)
```

grdHandle [in]

Тип: [Handle](#)

Хэндл, через который будет выполнена данная операция

loginFlags [in]

Тип: [GrdLM](#)

Флаг, устанавливающий режим лицензирования при использовании сетевых ключей. Для локальных ключей этот параметр игнорируется.

moduleLMS [in]

Тип: int

Номер модуля в таблице LMS, на котором будет регистрироваться копия приложения. Если LMS не используется, параметр должен быть равен 0xFFFFFFFF. Для локальных ключей этот параметр игнорируется

[Набор ошибок Guardant API](#)

Метод **GrdLogin** необходим для регистрации данной копии защищенного приложения на сервере Guardant Net или на локальном ключе.

Регистрация осуществляется в соответствии с критериями поиска, установленными методом [GrdSetFindMode](#).

Если критериям поиска удовлетворяют несколько обнаруженных ключей, регистрация производится на первом ответившем из них. Для того, чтобы избегать таких ситуаций, рекомендуется однозначно задавать параметры поиска ключа через [GrdSetFindMode](#). В противном случае Вам придется при помощи последовательного поиска ключей методом [GrdFind](#) построить список ключей, а затем методом [GrdSetFindMode](#) последовательно устанавливать в качестве критерия ID нужного ключа, выполнять к нему **GrdLogin**, проверять тот ли это ключ и если нет, то вызывать [GrdLogout](#) и так далее по списку. Это позволит избежать ошибочного выбора ключа при регистрации, но будет исполняться гораздо медленнее.

Поэтому правильный путь - делать так чтобы ключи приложений отличались в том числе и какими-либо параметрами, которые можно задать через [GrdSetFindMode](#). Например разные приложения разными номерами программ **prog**, разные версии приложений полями **ver** и так далее.

Для сетевых ключей

В соответствии с параметрами, установленными в конфигурационном файле [сетевых настроек клиента](#) функция **GrdLogin** производит поиск доступных серверов Guardant Net, на которых зарегистрированы удаленные ключи, удовлетворяющие критериям поиска. Условия поиска ключа задаются методом [GrdSetFindMode](#). Если такой ключ найден (т. е. информация о нем доступна серверу Guardant Net), производится уменьшение на 1 значения его сетевого ресурса, после чего регистрация данной копии приложения будет считаться выполненной (будет возвращено [GrdE.OK](#)). В противном случае будет возвращен соответствующий код ошибки, и регистрация будет отменена.

Захват лицензии осуществляется в соответствии со значением параметра *loginFlags*. Если значение этого параметра установлено как [GrdLM.PerStation](#), то все копии приложения, запущенные на данной рабочей станции будут использовать одну лицензию. То есть при запуске второй и последующих копий счетчик лицензий уменьшаться не будет.

Если значение *loginFlags* установлено как [GrdLM.PerProcess](#), то лицензии будут распределяться процессам, или копиям защищенного приложения. То есть при запуске каждого нового процесса счетчик лицензий на сервере Guardant Net будет уменьшаться на 1.

loginFlags установленное как [GrdLM.PerHandle](#) означает, что лицензия выделяется на каждый хэндл, создаваемый Guardant API (и/или автозащитой). Если при защите приложения используется несколько экземпляров библиотеки Guardant API и, соответственно несколько хэндлов (к примеру, при комбинированном использовании Guardant API и автозащиты), то лицензии будут выделены по числу хэндлов. В случае работы приложения с единственным хэндлом (т. е. защита только Guardant API или только автоматическая) распределение по хэндлам не будет отличаться от распределения по процессам.

При использовании License Management System и при наличии в ключе таблицы лицензий при регистрации кроме получения лицензии из общего счетчика захватывается лицензия из таблицы лицензий. Номер ячейки таблицы лицензий (номер модуля) должен быть указан в параметре *moduleLMS*. Если LMS не используется, значение параметра должно быть 0xFFFFFFFF.

Процедура регистрации будет выполнена успешно, если сервер Guardant Net располагает свободными лицензиями, а при использовании LMS - также и свободными лицензиями для конкретного модуля.

Сервер Guardant Net будет хранить нужную ему информацию о зарегистрированной копии защищенного приложения до момента снятия ее регистрации с помощью метода [GrdLogout](#), а все методы Guardant API для установки связи с сервером будут пользоваться его сетевым адресом, полученным при регистрации. Это существенно ускорит сетевой обмен.

Для локальных ключей

При работе с локальными ключами параметры *moduleLMS* и *loginFlags* игнорируются, поскольку локальные ключи не поддерживают распределения лицензий.

Внимание!

1. Вызов метода **GrdLogin** открывает сеанс непосредственного общения с ключом. Этот метод необходимо вызывать перед любым другим методом, который обращается непосредственно к ключу. При обращении к ключу до успешного выполнения **GrdLogin** будет возвращена ошибка [GrdE.NeedLogin](#).
2. Нельзя вызывать **GrdLogin** несколько раз в течение одного сеанса работы с ключом Guardant (т. е. до вызова метода [GrdLogout](#), который закрывает сеанс работы с ключом). При повторной попытке вызвать **GrdLogin** будет возвращена ошибка [GrdE.NeedLogout](#).
3. Не следует выполнять **GrdLogin** и [GrdLogout](#) без необходимости, это замедляет работу защищенного приложения. Рекомендуется придерживаться следующего порядка работы: приложение стартует (начало сессии работы с приложением) - вызов **GrdLogin**, приложение завершается (конец сессии работы с приложением) - вызов [Logout](#).

Java

```
public static GrdE GrdLogin(Handle grdHandle, GrdLM loginFlags)
public static GrdE GrdLogin(Handle grdHandle, int moduleLMS, GrdLM loginFlags)
```

grdHandle [in]

Тип: [Handle](#)

Хэндл, через который будет выполнена данная операция

loginFlags [in]

Тип: [GrdLM](#)

Флаг, устанавливающий режим лицензирования при использовании сетевых ключей. Для локальных ключей этот параметр игнорируется.

moduleLMS [in]

Тип: int

Номер модуля в таблице LMS, на котором будет регистрироваться копия приложения. Если LMS не используется, параметр должен быть равен 0xFFFFFFFF. Для локальных ключей этот параметр игнорируется

[Набор ошибок Guardant API](#)

Метод **GrdLogin** необходим для регистрации данной копии защищенного приложения на сервере Guardant Net или на локальном ключе.

Регистрация осуществляется в соответствии с критериями поиска, установленными методом [GrdSetFindMode](#).

Если критериям поиска удовлетворяют несколько обнаруженных ключей, регистрация производится на первом ответившем из них. Для того, чтобы избежать таких ситуаций, рекомендуется однозначно задавать параметры поиска ключа через [GrdSetFindMode](#). В противном случае Вам придется при помощи последовательного поиска ключей методом [GrdFind](#) построить список ключей, а затем методом [GrdSetFindMode](#) последовательно устанавливать в качестве критерия ID нужного ключа, выполнять к нему **GrdLogin**, проверять тот ли это ключ и если нет, то вызывать [GrdLogout](#) и так далее по списку. Это позволит избежать ошибочного выбора ключа при регистрации, но будет исполняться гораздо медленнее.

Поэтому правильный путь - делать так чтобы ключи приложений отличались в том числе и какими-либо параметрами, которые можно задать через [GrdSetFindMode](#). Например разные приложения разными номерами программ **prog**, разные версии приложений полями **ver** и так далее.

Для сетевых ключей

В соответствии с параметрами, установленными в конфигурационном файле [сетевых настроек клиента](#) функция **GrdLogin** производит поиск доступных серверов Guardant Net, на которых зарегистрированы удаленные ключи, удовлетворяющие критериям поиска. Условия поиска ключа задаются методом [GrdSetFindMode](#). Если такой ключ найден (т. е. информация о нем доступна серверу Guardant Net), производится уменьшение на 1 значения его сетевого ресурса, после чего регистрация данной копии приложения будет считаться выполненной (будет возвращено [GrdE.OK](#)). В противном случае будет возвращен соответствующий код ошибки, и регистрация будет отменена.

Захват лицензии осуществляется в соответствии со значением параметра *loginFlags*. Если значение этого параметра установлено как [GrdLM.PerStation](#), то все копии приложения, запущенные на данной рабочей станции будут использовать одну лицензию. То есть при запуске второй и последующих копий счетчик лицензий уменьшаться не будет.

Если значение *loginFlags* установлено как [GrdLM.PerProcess](#), то лицензии будут распределяться процессам, или копиям защищенного приложения. То есть при запуске каждого нового процесса счетчик лицензий на сервере Guardant Net будет уменьшаться на 1.

loginFlags установленное как [GrdLM.PerHandle](#) означает, что лицензия выделяется на каждый хэндл, создаваемый Guardant API (и/или автозащитой). Если при защите приложения используется несколько экземпляров библиотеки Guardant API и, соответственно несколько хэндлов (к примеру, при комбинированном использовании Guardant API и автозащиты), то лицензии будут выделены по числу хэндлов. В случае работы приложения с единственным хэндлом (т. е. защита только Guardant API или только автоматическая) распределение по хэндлам не будет отличаться от распределения по процессам.

При использовании License Management System и при наличии в ключе таблицы лицензий при регистрации кроме получения лицензии из общего счетчика захватывается лицензия из таблицы лицензий. Номер ячейки таблицы лицензий (номер модуля) должен быть указан в параметре *moduleLMS*. Если LMS не используется, значение параметра должно быть 0xFFFFFFFF.

Процедура регистрации будет выполнена успешно, если сервер Guardant Net располагает свободными лицензиями, а при использовании LMS - также и свободными лицензиями для конкретного модуля.

Сервер Guardant Net будет хранить нужную ему информацию о зарегистрированной копии защищенного приложения до момента снятия ее регистрации с помощью метода [GrdLogout](#), а все методы Guardant API для установки связи с сервером будут пользоваться его сетевым адресом, полученным при регистрации. Это существенно ускорит сетевой обмен.

Для локальных ключей

При работе с локальными ключами параметры *moduleLMS* и *loginFlags* игнорируются, поскольку локальные ключи не поддерживают распределения лицензий.

Внимание!

1. Вызов метода **GrdLogin** открывает сеанс непосредственного общения с ключом. Этот метод необходимо вызывать перед любым другим методом, который обращается непосредственно к ключу. При обращении к ключу до успешного выполнения **GrdLogin** будет возвращена ошибка [GrdE.NeedLogin](#).
2. Нельзя вызывать **GrdLogin** несколько раз в течение одного сеанса работы с ключом Guardant (т. е. до вызова метода [GrdLogout](#), который закрывает сеанс работы с ключом). При повторной попытке вызвать **GrdLogin** будет возвращена ошибка [GrdE.NeedLogout](#).
3. Не следует выполнять **GrdLogin** и [GrdLogout](#) без необходимости, это замедляет работу защищенного приложения. Рекомендуется придерживаться следующего порядка работы: приложение стартует (начало сессии работы с приложением) - вызов **GrdLogin**, приложение завершается (конец сессии работы с приложением) - вызов [Logout](#).

