

gsq/cmd

Утилита командной строки для работы с базами данных

Версия 4.1 от 3 февраля 2017 г.

Руководство пользователя

Содержание

Содержание	2
Введение	4
Введение	4
История изменений	5
Сравнение редакций	6
Системные требования	7
Лицензионное соглашение	8
Использование gsqlcmd	9
Использование gsqlcmd	9
Работа с подключениями	9
Выполнение SQL-скриптов	9
Экспорт данных из базы данных	10
Использование переменных в SQL скриптах	10
Команды вывода сообщений без обращения к серверу баз данных	10
Транзакционный режим выполнения	10
Режим трассировки	10
Импорт данных CSV в базы данных	11
Формат командной строки	16
Режимы работы командной строки	17
Execute	17
Prepare	17
Parse	17
Insert, Update, Delete	17
Merge	17
Fmt+	17
Fmt	18
Create	18
Ace	18
Bulk	18
Ini	18
LocalConfig UserConfig AppConfig	18
Connections	18
Stamp	18
Help	18
Позиционные параметры	19
<Подключение>	19
<SQL-запрос> <SQL-файл>	19
<CSV-файл>	20
<Выходной файл>	20
Параметры выполнения запросов	21
/CommandTimeout=<количество секунд>	21
/ConnectionTimeout=<количество секунд>	21
/NoTransaction	21
/InputCodePage=<кодировка>	21
/OutputCodePage=<кодировка>	21
/Set=<параметр>=<значение функция>[:...]	21
Функции параметров опций /set и /add	22
/Trace	22
Общие выходные параметры	23
/AddRowNum	23
/Append	23
/AsText AsCSV AsHTML	23
/DateTimeFormat=<формат даты>	23
/NoHeaders	23
Параметры CSV	24
/Add=<заголовок>=<значение функция>[:<разделитель>...]	24
Функции параметров опций /set и /add	24
/[output]Separator=<разделитель> tab	24
/inputSeparator=<разделитель> tab	24

/QuoteChar=<символ>	24
/EscapeChar=<символ>	24
Параметры вывода HTML	25
/NoTemplate	25
/Placeholder=<имя форматной вставки>	25
/Template=<файл шаблона HTML>	25
/Title=<наименование документа>	25
Параметры генерации кода	25
/Table=<таблица представление хранимая процедура> <файл шаблона SQL кода>	25
/fmt=<файл формата>	25
/InsertIdentity	25
/InsertNulls	25
/SingleLineSQL	25
/GroupSize=<количество строк>	25
/keys=<поле>[,<поле>[,...]]	25
/mssql sqlce mysql oracle db2 nuodb pgsql sqlite	26
Коды возврата	26
gConnectionManager	27
Файл конфигурации	29
Регистрация продукта	30
Выбор продукта	30
Выбор редакции	31
Ввод данных владельца лицензии	31
Регистрация онлайн	32
Регистрация по электронной почте	33
Техническая поддержка	35
Техническая поддержка	35
Вопросы и ответы	35

Введение

Утилита командной строки `gsqcmd` позволяет:

- Выполнять SQL команды и SQL скрипты с параметрами.
- Экспортировать данные из баз данных в текстовые файлы, CSV и HTML.
- Создавать файлы конфигурации для импорта CSV данных в базы данных.
- Генерировать команды INSERT, UPDATE, DELETE, MERGE по данным CSV.
- Импортировать данные CSV в базы данных.

Утилита позволяет единообразно работать со всеми поддерживаемыми серверами баз данных.

При этом не требуется наличие установленных утилит командной строки от производителя сервера баз данных.

Совет: дополнительная утилита `gwebcmd` позволяет загружать веб данные и конвертировать веб данные и текстовые файлы XML, HTML, CSV, JSON, Text в CSV.

Поддерживаются следующие сервера баз данных:

- Microsoft Azure SQL Database
- Microsoft SQL Server
- Microsoft SQL Server Compact
- Oracle Database
- IBM DB2
- MySQL и MariaDB
- NuoDB
- PostgreSQL
- SQLite

Поддерживаются следующие форматы экспорта данных:

- Текст
- CSV
- HTML

Основные разделы:

- [Использование gsqlcmd](#)
- [Импорт данных CSV в базы данных](#)
- [Сравнение редакций](#)
- [Командная строка](#)
- [Менеджер подключений](#)
- [Файл конфигурации](#)
- [Вопросы и ответы](#)

История изменений

Условные обозначения:

- [+] Новые возможности
- [-] Исправленные ошибки
- [*] Изменения

Версия 4.1 от 03.02.2017

- [-] Исправлена ошибка определения первичных ключей таблиц SQL Server в некоторых случаях. Ошибка была добавлена в gsqlcmd 4.0.

Версия 4.0.3 от 25.01.2017

- [-] Исправлена ошибка создания баз данных SQLite.

Версия 4.0 от 17.01.2017

- [+] Добавлена поддержка баз данных SQLite.
- [+] Добавлен режим CreateSQLite3 для создания новых файлов баз данных SQLite.
- [+] Добавлена опция keys для указания полей, которые следует использовать вместо полей первичных ключей при генерации команд INSERT, UPDATE, DELETE.
Это удобно использовать при слиянии данных в разных базах данных по суррогатным ключам, типа email или ИНН, вместо числовых первичных ключей.
- [+] Продукт можно приобрести по подписке.
- [*] Обновление на версию 4.0 с предыдущих версий можно выполнить бесплатно.
- [*] Значительно улучшена производительность выполнения команд MySQL.
- [*] Обновлен gConnectionManager 3.0.

Версия 3.4 от 02.11.2016

- [-] Пропуск команд SET IDENTITY_INSERT при генерации на основе данных CSV.
- [-] Пропуск пустых строк при создании хранимых процедур MySQL.

Версия 3.3 от 21.06.2016

- [+] Добавлено приложение для генерации скриптов установки приложений баз данных Microsoft SQL Server.
- [!] Новый веб-сайт продукта - www.gsqlcmd.com.
- [*] Обновлен gConnectionManager 2.2.

Версия 3.2 от 15.03.2016

- [+] Пакет со стандартным установщиком Windows.
- [*] Обновлен gConnectionManager 2.1.

Версия 3.1 от 23.12.2015

- [+] Возможность установки всех свойств строки подключения.
- [*] Обновлен провайдер PostgreSQL.
- [*] Обновлен провайдер MySQL.

Версия 3.0 от 20.10.2015

- [!] Изменено лицензионное соглашение - удалена бесплатная редакция gsqlcmd Express. Вы можете использовать версию без регистрации без ограничений по времени.

Сравнение редакций

gsqcmd позволяет выполнять следующие задачи:

1. Выполнение SQL запросов и скриптов с параметрами.
2. Экспорт данных из баз данных в CSV, HTML и текст.
3. Подготовка файлов конфигурации для импорта CSV данных в базы данных.
4. Генерация команд INSERT, UPDATE, DELETE, MERGE по данным CSV.
5. Импорт данных CSV в базы данных.

gsqcmd позволяет заменить утилиты sqlcmd, bcp, sqlplus, db2 и mysql во многих сценариях и с дополнительными возможностями.

Утилиты позволяют получать результаты с меньшими усилиями и единообразно для всех основных платформ баз данных.

Возможность	Free	Personal	Enterprise
Возможности, доступные на всех поддерживаемых платформах баз данных			
Выполнение SQL запросов и скриптов с параметрами	✓	✓	✓
Экспорт данных в CSV, HTML и текст	✓	✓	✓
Генерация SQL кода CREATE TABLE для последующего импорта CSV	✓	✓	✓
Генерация SQL кода INSERT, UPDATE, DELETE, MERGE для данных CSV	500 строк	✓	✓
Генерация SQL кода по SQL шаблону для данных CSV	500 строк	✓	✓
Генерация SQL кода вызова хранимых процедур для данных CSV	500 строк	✓	✓
Генерация файла schema.ini для импорта данных CSV через ODBC	✓	✓	✓
Именованные подключения, открытые и зашифрованные	✓	✓	✓
Поддерживаемые платформы баз данных: SQL Server, SQL Server Compact, MySQL, Oracle, DB2, NuoDB, PostgreSQL, SQLite	✓	✓	✓
Возможности, доступные для Microsoft SQL Server			
Генерация файлов форматов и SQL кода OPENROWSET BULK	✓	✓	✓
Генерация SQL кода OPENROWSET для MICROSOFT.ACE.OLEDB.12.0	✓	✓	✓
Лицензирование			
Коммерческое использование	✓	✗	✓

Системные требования

Поддерживаемые архитектуры:

- x86
- x64

Поддерживаемые операционные системы:

- Windows XP SP3, Vista SP1, 7, 8, 8.1, 10
- Windows Server 2003 SP3, 2008, 2008 R2, 2012, 2012 R2, 2016

Поддерживаемые версии Microsoft SQL Server:

- Microsoft SQL Server 2000
- Microsoft SQL Server 2005, 2008, 2008 R2
- Microsoft SQL Server 2012, 2014, 2016, включая Express LocalDB
- Microsoft Azure SQL Database

Поддерживаемые версии Microsoft SQL Server Compact:

- Microsoft SQL Server Compact 3.5, 4.0

Поддерживаемые версии Oracle Database:

- Oracle Database 10g Release 1, Release 2
- Oracle Database 11g Release 1, Release 2
- Oracle Database 12c Release 1

Поддерживаемые версии IBM DB2:

- IBM DB2 9.5, 9.7, 9.8, 10.1, 10.5, 11.1

Требуется установленный IBM DB2 .NET Provider, IBM DB2 OLE DB Provider или IBM DB2 ODBC драйвер.

Поддерживаемые версии Oracle MySQL:

- MySQL 5.0, 5.1, 5.2, 5.5, 5.6, 5.7

Поддерживаемые версии SkySQL MariaDB:

- MariaDB 5.1, 5.2, 5.3, 5.5, 10.0, 10.1, 10.2

Поддерживаемые версии NuoDB:

- NuoDB 2.0.4, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6

Поддерживаемые версии PostgreSQL:

- PostgreSQL 8.0, 8.1, 8.2, 8.3, 8.4, 9.0, 9.1, 9.2, 9.3, 9.4

Поддерживаемые версии SQLite:

- SQLite 2, 3

Лицензионное соглашение

Программа gsqlcmd, далее gsqlcmd, распространяется в соответствии со следующими положениями:

1. Всеми авторскими правами на gsqlcmd эксклюзивно владеет Закрытое акционерное общество "Гартл Технолоджи Корпорейшн".
2. Вам предоставляется неэксклюзивная лицензия на использование gsqlcmd на одном компьютере. Бесплатная версия имеет некоторые ограничения. Для снятия ограничений следует приобрести лицензию и зарегистрировать редакцию gsqlcmd Personal или gsqlcmd Enterprise. Использование редакции Personal для коммерческих целей, например юридическими лицами, запрещено.
3. Незарегистрированная программа gsqlcmd может распространяться свободно, кроме случаев, оговорённых ниже, лишь в том виде, в котором она поставляется, т.е. без всяких изменений.
 - a. Никакие физические или юридические лица не имеют права распространять отдельные части пакета без письменного разрешения владельца авторских прав.
 - b. Запрещается распространять программу gsqlcmd в комплекте других программных пакетов без письменного разрешения владельца авторских прав.
 - c. Запрещается распространять программу gsqlcmd на одном носителе и/или в одном пакете программ вместе с пиратскими утилитами для "взлома", ключевыми файлами и генераторами ключей.
4. ПРОГРАММА gsqlcmd РАСПРОСТРАНЯЕТСЯ ПО ПРИНЦИПУ "КАК ЕСТЬ". ПРИ ЭТОМ НЕ ПРЕДУСМАТРИВАЕТСЯ НИКАКИХ ГАРАНТИЙ, ЯВНЫХ ИЛИ ПОДРАЗУМЕВАЕМЫХ. ВЫ ИСПОЛЬЗУЕТЕ ЕЕ НА СВОЙ СОБСТВЕННЫЙ РИСК. НИ ВЛАДЕЛЕЦ АВТОРСКИХ ПРАВ, НИ ЕГО УПОЛНОМОЧЕННЫЕ АГЕНТЫ НЕ НЕСУТ ОТВЕТСТВЕННОСТИ ЗА ПОТЕРИ ДАННЫХ, ПОВРЕЖДЕНИЯ, ПОТЕРИ ПРИБЫЛИ ИЛИ ЛЮБЫЕ ДРУГИЕ ВИДЫ ПОТЕРЬ, СВЯЗАННЫЕ С ИСПОЛЬЗОВАНИЕМ (ПРАВИЛЬНЫМ ИЛИ НЕПРАВИЛЬНЫМ) ЭТОЙ ПРОГРАММЫ.
5. Вы не можете использовать, копировать, эмулировать, клонировать, сдавать в аренду, давать напрокат, продавать, изменять, декомпилировать, дизассемблировать, передавать программу или её часть иначе, чем это описано в данной лицензии. Любое подобное неавторизованное использование приводит к немедленному и автоматическому прекращению действия этой лицензии и может повлечь за собой уголовное и/или гражданское преследование. Все права, явно не предоставленные здесь, принадлежат Закрытому акционерному обществу "Гартл Технолоджи Корпорейшн".
6. Установка и использование программы gsqlcmd свидетельствует о вашем согласии с условиями данной лицензии.
7. Если вы не согласны с условиями данной лицензии, то должны удалить файлы gsqlcmd со своих устройств хранения информации и отказаться от дальнейшего использования этого программного продукта.

Использование gsqlcmd

Утилита gsqlcmd позволяет решать следующие основные задачи:

- Выполнение SQL-скриптов для создания, изменения или удаления объектов базы данных (DDL) и данных (DML).
- Экспорт данных в текст, CSV или HTML.
- Подготовка файлов форматов и SQL кода для импорта данных CSV в базы данных.
- Импорт данных CSV в базы данных.

Эти задачи выполняются более просто, чем при использовании утилит командной строки от производителей баз данных, и, в дополнение, имеют единый синтаксис для всех серверов баз данных.

Совет: дополнительная утилита gwebcmd позволяет загружать веб данные и конвертировать веб данные и текстовые файлы XML, HTML, CSV, JSON, Text в CSV.

Форматы командной строки:

```
gsqlcmd [execute] <подключение> <sql-запрос> [<выходной файл>] [<опции>]
gsqlcmd [execute] <подключение> <sql-файл> [<выходной файл>] [<опции>]

gsqlcmd insert <подключение> <csv-файл> [<выходной файл>] [<опции>]
gsqlcmd merge <подключение> <csv-файл> [<выходной файл>] [<опции>]

gsqlcmd fmt+ <csv-файл> [<опции>]
gsqlcmd fmt <csv-файл> [<выходной файл>] [<опции>]
gsqlcmd create|bulk|ini|ace <csv-файл> [<выходной файл>] [<опции>]

gsqlcmd localconfig | userconfig | appconfig
gsqlcmd connections

gsqlcmd help [chm]
```

См. описание режимов и параметров в разделе [Командная строка](#).

При использовании в командных файлах можно использовать [коды возврата](#).

Ниже описаны варианты использования и полезные задачи.

Работа с подключениями

При выполнении команд или скриптов требуются данные подключения.

Данные подключения можно указать в первом позиционном параметре при вызове или в [конфигурационном файле](#).

В последнем случае, в качестве параметра можно указать только имя подключения, например mssql-rtd.

Строка подключения ищется по имени в трех файлах в указанном порядке:

1. gsqlcmd.exe.config в текущем каталоге.
2. gsqlcmd.exe.config в каталоге %LocalAppData%\Gartle\gsqlcmd
3. gsqlcmd.exe.config в каталоге утилиты.

Для редактирования файлов в визуальном режиме можно использовать продукт [gConnectionManager](#), который устанавливается вместе с gsqlcmd.

Для быстрого вызова редактора можно использовать команды: localconfig, userconfig или appconfig, которые открывают соответствующий файл.

Для вывода имен всех подключений можно использовать режим connections.

Выполнение SQL-скриптов

Пример выполнения скрипта создания объектов базы данных rtd-setup-sqlce.sql для предопределенного подключения sqlce-rtd:

```
gsqlcmd sqlce-rtd rtd-setup-sqlce.sql
```

Пример выполнения скрипта создания объектов базы данных rtd-setup-sqlce.sql в базе данных rtd.sdf Microsoft SQL Server Compact:

```
gsqlcmd "Data Source=rtd.sdf" rtd-setup-sqlce.sql
```

Пример выполнения скрипта создания базы данных rtd-setup-mssql.sql в базе данных RTD Microsoft SQL Server:

```
gsqlcmd "Provider=SQLOLEDB;Data Source=.\SQLEXPRESS;Integrated Security=SSPI;Initial Catalog=RTD" rtd-setup-mssql.sql
```

Экспорт данных из базы данных

Пример вывода результата запроса командной строки на консоль:

```
gsqcmd mysql-rtd "SELECT * FROM INFORMATION_SCHEMA.TABLES"
```

По умолчанию, при выводе на консоль и в файлы с расширением *.txt используется вывод в виде текста.

Пример вывода запроса в файле export-quotes.sql в файл quotes.csv в формате CSV:

```
gsqcmd db2-rtd export-quotes.sql quotes.csv
```

В файлы с расширением *.csv данные по умолчанию выводятся в формате CSV.

Если используется перенаправление вывода и требуется формат CSV, то его требуется указать. Например:

```
gsqcmd rtd-db2 export-quotes.sql /asCSV > quotes.csv
```

Также можно задать дополнительные параметры для вывода:

```
gsqcmd rtd-db2 export-quotes.sql quotes.csv /separator=, /datetimeformat=yyyy-MM-dd
```

Пример вывода результатов в формат HTML:

```
gsqcmd rtd-ora export-quotes.sql quotes.htm
```

В примере используется встроенный шаблон страницы HTML, который используется по умолчанию.

Шаблон можно задать в параметрах команды. Например:

```
gsqcmd rtd-ora export-quotes.sql quotes.htm /template=table-template.htm
```

Шаблон можно отключить опцией /notemplate.

Использование переменных в SQL скриптах

SQL команды и скрипты могут использовать значения переменных, переданные в командной строке в опции /set.

Например:

```
gsqcmd mssql-rtd "SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = @Table" /set=Table=RealTimeTables
```

```
gsqcmd mysql-rtd "SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = :Table" /set=Table=real_time_tables
```

Для Microsoft SQL Server, Microsoft SQL Server Compact и SQLite используйте формат @Переменная, для MySQL, MariaDB, Oracle Database, IBM DB2, NuoDB и PostgreSQL используйте :Переменная.

Переменные не следует использовать со скриптами, которые содержат объявления переменных, которые необходимо передать серверу,

например в командах создания хранимых процедур и триггеров.

Команды вывода сообщений без обращения к серверу баз данных

SQL-скрипты могут содержать команду -- print <сообщение> с начала строки, которая обрабатывается без обращения к серверу баз данных.

Это удобно использовать для отслеживания прогресса выполнения скриптов.

Данная техника единообразно работает для всех серверов баз данных и позволяет выводить сообщения, даже если это не поддерживается сервером баз данных.

Пример:

```
-- print Таблица ColumnTranslation создана
```

Транзакционный режим выполнения

SQL команды и скрипты по умолчанию выполняются в транзакционном режиме с уровнем изоляции ReadCommitted.

При этом выполняются либо все команды, либо ни одна.

Для отключения режима можно использовать опцию /notransaction.

Режим трассировки

Если скрипт не выполняется, или выполняется долго, то можно включить режим трассировки опцией /trace.

По умолчанию, данные трассировки выводятся на консоль. Вывод можно переопределить в файле gsqlcmd.exe.config.

Импорт данных CSV в базы данных

gsqcmd позволяет генерировать следующие файлы для импорта данных CSV в базы данных:

- SQL код CREATE TABLE.
- Файлы форматов для функции OPENROWSET(BUCK...) и утилиты bcp (SQL Server).
- INI файлы для драйверов ODBC.
- SQL код INSERT INTO ... SELECT для функции OPENROWSET(BUCK...) (SQL Server).
- SQL код INSERT INTO ... SELECT для функции OPENROWSET и MICROSOFT.ACE.OLEDB.12.0 (SQL Server).
- Команды INSERT.
- Команды INSERT и UPDATE.

Совет: вы можете преобразовать XML, HTML, JSON и текст в CSV с помощью утилиты gwebcmd.

Создание файлов форматов

Файлы форматов используются для импорта данных в Microsoft SQL Server с помощью утилиты bcp и функции OPENROWSET(BUCK...).

gsqcmd позволяет генерировать такие файлы и использует их как источник наименований колонок.

Соответственно, вы можете сгенерировать файл формата, отредактировать его и генерировать остальные файлы на его основе.

К примеру, исходный CSV файл payments.csv содержит следующие данные:

```
ID;Date;Sum;"Account Name";"Company Name";"Item Name";"Comment"
1;01/10/2014 00:00:00;200000;"My Bank";"Rose, Inc";"Revenue";""
2;01/10/2014 00:00:00;-50000;"My Bank";"Land, Inc";"Expenses";""
```

Вы можете сгенерировать файл формата с использованием режима fmt:

```
gsqcmd fmt payments.csv payments.fmt
```

и отредактировать наименования колонок (например, убрать пробелы):

```
9.0
7
1  SQLCHAR  0  255  ";"          1  "ID"          ""
2  SQLCHAR  0  255  ";"          2  "Date"        ""
3  SQLCHAR  0  255  ";\\""      3  "Sum"         ""
4  SQLCHAR  0  255  "\"";\""    4  "AccountName" ""
5  SQLCHAR  0  255  "\"";\""    5  "CompanyName" ""
6  SQLCHAR  0  255  "\"";\""    6  "ItemName"    ""
7  SQLCHAR  0  255  "\"\r\n"   7  "Comment"     ""
```

Совет: вы можете отменить импорт определенных колонок. Для этого следует заменить индекс колонки (колонка 6) на 0.

Создание таблиц в базе данных

Далее вы можете создать SQL код для создания таблицы CREATE TABLE:

```
gsqcmd create payments.csv payments.create.sql /table=dbo.Payment /fmt=payments.fmt
```

Вы можете задать целевую платформу базы данных используя опции: /mssql, /sqlce, /mysql, /oracle, /db2, /nuodb, /pgsql.

Результат файла payments.create.sql:

```
CREATE TABLE [dbo].[Payment] (
    [ID] int IDENTITY(1,1) NOT NULL,
    [Date] datetime NULL,
    [Sum] int NULL,
    [AccountName] nvarchar(255) NULL,
    [CompanyName] nvarchar(255) NULL,
    [ItemName] nvarchar(255) NULL,
    [Comment] nvarchar(255) NULL,
    CONSTRAINT [PK_Payment_dbo] PRIMARY KEY CLUSTERED (
        [ID] ASC
    )
)
GO
print 'Table [dbo].[Payment] has been created'
GO
```

Проверьте и измените типы данных по необходимости. Например, поле [Sum] получило тип int, т.к. исходный набор содержал только целые числа.

Вы можете изменить тип int на money или float, например.

Вы можете выполнить код CREATE TABLE в режиме execute:

```
gsqcmd execute <подключение> payments.create.sql
```

Совет: вы можете создавать именованные строки подключения в визуальном режиме.
Используйте одну из команд:

```
gsqcmd appconfig  
gsqcmd userconfig  
gsqcmd localconfig
```

Вывести список всех определенных именованных подключений можно командой:

```
gsqcmd connections
```

Импорт данных CSV в Microsoft SQL Server с использованием утилиты BCP

Вы можете импортировать данные CSV в Microsoft SQL Server с использованием утилиты bcp, входящей в состав Microsoft SQL Server.

К примеру, вы можете импортировать данные CSV из файла payments.csv в созданную таблицу dbo.Payment используя команду:

```
bcp "dbo.Payment" in payments.csv -S . -d <database> -T -f payments.fmt -E
```

Опция -E используется для вставки значений identity.

Совет: утилита bcp не поддерживает кодировку UTF-8.

Вы можете конвертировать файл CSV в кодировку Windows ANSI или использовать методы, описанные ниже.

См. описание утилиты bcp на странице <http://msdn.microsoft.com/ru-ru/library/ms162802.aspx>.

Использование функции OPENROWSET(BULK...)

Для генерации SQL кода INSERT INTO ... SELECT используйте следующую команду:

```
gsqcmd bulk payments.csv payments.insert.bulk.sql /table=dbo.Payment /fmt=payments.fmt /insertIdentity
```

Результат файла:

```
SET IDENTITY_INSERT dbo.Payment ON;  
INSERT INTO dbo.Payment  
    (  
        [ID]  
        , [Date]  
        , [Sum]  
        , [AccountName]  
        , [CompanyName]  
        , [ItemName]  
        , [Comment]  
    )  
SELECT  
    t.[ID]  
    , t.[Date]  
    , t.[Sum]  
    , t.[AccountName]  
    , t.[CompanyName]  
    , t.[ItemName]  
    , t.[Comment]  
FROM  
    (  
        SELECT  
            [ID] AS [ID]  
            , [Date] AS [Date]  
            , [Sum] AS [Sum]  
            , [AccountName] AS [AccountName]  
            , [CompanyName] AS [CompanyName]  
            , [ItemName] AS [ItemName]  
            , [Comment] AS [Comment]  
        FROM  
            OPENROWSET(  
                BULK 'D:\payments.csv',  
                FORMATFILE = 'D:\payments.fmt',  
                CODEPAGE = '1251',  
                FIRSTROW = 2) t  
    ) t  
SET IDENTITY_INSERT dbo.Payment OFF;
```

Если не указать опцию /table, то заголовок INSERT INTO будет пропущен.

Обратите внимание на опцию /insertIdentity. Она определяет команды SET IDENTITY_INSERT для вставки значений полей identity из исходного файла CSV.

В качестве альтернативы, поле можно пропустить, если в колонке 6 файла формата задать 0 для поля.

Обратите внимание, что функция OPENROWSET требует указания абсолютных путей к файлам CSV и формата.
Вы можете отредактировать реальные пути к файлам.

Вы можете импортировать данные, используя этот файл, если пути к файлам доступны вашему экземпляру SQL Server.

Вы можете выполнить этот код в SQL Server Management Studio, с помощью утилиты sqlcmd пакета SQL Server или с помощью gsqlcmd:

```
gsqlcmd execute <подключение> payments.insert.bulk.sql
```

Также вы можете вставить этот код в хранимые процедуры.

Вы можете проверить загруженные данные, используя команду SELECT в командной строке. Например:

```
gsqlcmd execute <подключение> "SELECT * FROM dbo.Payment"
```

Пример:

ID	Date	Sum	AccountName	CompanyName	ItemName	Comment
1	01/10/2014 00:00:00	200000	My Bank	Rose, Inc	Revenue	
2	01/10/2014 00:00:00	-50000	My Bank	Land, Inc	Expenses	

Создание секций файла SCHEMA.INI для импорта с использованием ODBC

Вы можете легко создавать секции для файла schema.ini для последующего импорта с использованием ODBC.

Например, выполните:

```
gsqlcmd ini payments.csv payments.ini /fmt=payments.fmt
```

Результат файла payments.ini:

```
[payments.csv]
ColNameHeader=True
Format=Delimited(;)
MaxScanRows=100
CharacterSet=1250
Col1="ID" Char Width 255
Col2="Date" Char Width 255
Col3="Sum" Char Width 255
Col4="AccountName" Char Width 255
Col5="CompanyName" Char Width 255
Col6="ItemName" Char Width 255
Col7="Comment" Char Width 255
```

Далее следует скопировать и добавить строки в файл schema.ini.

Использование функции OPENROWSET с MICROSOFT.ACE.OLEDB.12.0

Утилита bcp и функция OPENROWSET(BUCK...) не поддерживают кодировку UTF-8.

Одним из решений может быть использование провайдера MICROSOFT.ACE.OLEDB.12.0 в функции OPENROWSET.

Вы можете загрузить провайдер MICROSOFT.ACE.OLEDB.12.0 со страницы

<http://www.microsoft.com/en-us/download/details.aspx?id=13255>

Для генерации SQL кода используйте следующую команду:

```
gsqlcmd ace payments.csv payments.insert.ace.sql /table=dbo.Payment /fmt=payments.fmt
```

Результат файла payments.insert.ace.sql:

```
INSERT INTO dbo.Payment
(
    [ID]
    , [Date]
    , [Sum]
    , [AccountName]
    , [CompanyName]
    , [ItemName]
    , [Comment]
)
SELECT
    t.[ID]
    , t.[Date]
    , t.[Sum]
    , t.[AccountName]
    , t.[CompanyName]
    , t.[ItemName]
    , t.[Comment]
FROM
(
    SELECT
        [ID] AS [ID]
        , [Date] AS [Date]
        , [Sum] AS [Sum]
        , [AccountName] AS [AccountName]
```

```

, [CompanyName] AS [CompanyName]
, [ItemName] AS [ItemName]
, [Comment] AS [Comment]
FROM
OPENROWSET ('MICROSOFT.ACE.OLEDB.12.0',
            'Text;Database=D:\;HDR=Yes;Format=Delimited(;)',
            'SELECT * FROM [payments.csv]') t
) t

```

Перед использованием этого кода следует добавить секцию [payments.csv] в файл schema.ini. См. предыдущий раздел.

Вы можете выполнить этот код в SQL Server Management Studio, используя утилиту sqlcmd от Microsoft SQL Server или утилиту gsqlcmd:

```
gsqlcmd execute <подключение> payments.insert.ace.sql
```

Также вы можете вставить этот код в хранимую процедуру.

Создание команд INSERT для импорта данных CSV

Вы можете сгенерировать команды INSERT для вставки данных CSV и выполнить их в вашей интегрированной среде разработки, используя утилиты командной строки от производителя баз данных или утилиту gsqlcmd.

Например:

```
gsqlcmd insert <подключение> payments.csv payments.insert.sql /table=dbo.Payment /fmt=payments.fmt
```

Результат файла payments.insert.sql:

```

INSERT INTO [dbo].[Payment] ([Date], [Sum], [AccountName], [CompanyName], [ItemName])
VALUES ('20140110 00:00:00.000', 200000, N'My Bank', N'Rose, Inc', N'Revenue');
INSERT INTO [dbo].[Payment] ([Date], [Sum], [AccountName], [CompanyName], [ItemName])
VALUES ('20140110 00:00:00.000', -50000, N'My Bank', N'Land, Inc', N'Expenses');

```

Режим insert требует подключение в качестве первого параметра и параметр /table.

gsqlcmd загружает определение таблицы и генерирует код, соответствующий целевой платформе баз данных.

Вы можете задать ключ /insertIdentity для вставки значений полей identity из исходного файла CSV.

Сгенерированный код легко выполнить с помощью утилиты gsqlcmd:

```
gsqlcmd execute <подключение> payments.insert.sql
```

Бесплатная версия позволяет генерировать команды INSERT для файлов CSV не более 500 строк.

Вы можете приобрести Personal или Enterprise Edition для снятия данного ограничения.

Создание команд INSERT и UPDATE для слияния данных CSV

Вы можете сгенерировать команды INSERT и UPDATE для слияния данных CSV и целевой таблицы.

Например:

```
gsqlcmd merge AzureDemo payments.csv payments.merge.sql /table=dbo.Payment /fmt=payments.fmt
```

Результат файла payments.merge.sql:

```

UPDATE [dbo].[Payment]
SET
    [Date] = s.[Date]
    , [Sum] = s.[Sum]
    , [AccountName] = s.[AccountName]
    , [CompanyName] = s.[CompanyName]
    , [ItemName] = s.[ItemName]
    , [Comment] = s.[Comment]
FROM
[dbo].[Payment] t INNER JOIN (
    SELECT 1 AS [ID], '20140110 00:00:00.000' AS [Date], 200000 AS [Sum], N'My Bank' AS [AccountName],
           N'Rose, Inc' AS [CompanyName], N'Revenue' AS [ItemName], NULL AS [Comment]
    UNION ALL SELECT 2 AS [ID], '20140110 00:00:00.000' AS [Date], -50000 AS [Sum], N'My Bank' AS [AccountName],
           N'Land, Inc' AS [CompanyName], N'Expenses' AS [ItemName], NULL AS [Comment]
) s ON t.[ID] = s.[ID];

INSERT INTO [dbo].[Payment] ([Date], [Sum], [AccountName], [CompanyName], [ItemName], [Comment]) SELECT s.[Date],
    s.[Sum], s.[AccountName], s.[CompanyName], s.[ItemName], s.[Comment]
FROM (
    SELECT s.[ID], s.[Date], s.[Sum], s.[AccountName], s.[CompanyName], s.[ItemName], s.[Comment]
    FROM (
        SELECT 1 AS [ID], '20140110 00:00:00.000' AS [Date], 200000 AS [Sum], N'My Bank' AS [AccountName],
              N'Rose, Inc' AS [CompanyName], N'Revenue' AS [ItemName], NULL AS [Comment]
        UNION ALL SELECT 2 AS [ID], '20140110 00:00:00.000' AS [Date], -50000 AS [Sum], N'My Bank' AS [AccountName],
              N'Land, Inc' AS [CompanyName], N'Expenses' AS [ItemName], NULL AS [Comment]
    ) s
) s

```

```
        LEFT OUTER JOIN [dbo].[Payment] t ON t.[ID] = s.[ID] WHERE t.[ID] IS NULL
    ) s;
GO
print 'Processed 2 total records';
```

Режим merge требует подключение в качестве первого параметра и параметр /table.
gsqcmd загружает определение таблицы и генерирует код, соответствующий целевой платформе баз данных.

Вы можете задать ключ /insertIdentity для вставки значений полей identity из исходного файла CSV.

Сгенерированный код легко выполнить с помощью утилиты gsqlcmd:

```
gsqcmd execute <подключение> payments.merge.sql
```

Бесплатная версия позволяет генерировать команды INSERT и UPDATE для файлов CSV не более 500 строк.
Вы можете приобрести Personal или Enterprise Edition для снятия данного ограничения.

Формат командной строки

Режимы командной строки:

```
gsqcmd [execute] <подключение> <sql-запрос | sql-файл> [<выходной файл> [<опции>]
gsqcmd prepare <подключение> <sql-запрос | sql-файл> [<выходной файл> [<опции>]
gsqcmd parse <подключение> <sql-запрос | sql-файл> [<выходной файл> [<опции>]

gsqcmd insert <подключение> <csv-файл> [<выходной файл> [<опции>]
gsqcmd update <подключение> <csv-файл> [<выходной файл> [<опции>]
gsqcmd delete <подключение> <csv-файл> [<выходной файл> [<опции>]
gsqcmd merge <подключение> <csv-файл> [<выходной файл> [<опции>]

gsqcmd fmt+ <csv-файл> [<опции>]
gsqcmd fmt <csv-файл> [<выходной файл> [<опции>]
gsqcmd create|bulk|ini|ace <csv-файл> [<выходной файл> [<опции>]

gsqcmd localconfig | userconfig | appconfig
gsqcmd connections

gsqcmd stamp [<формат datetime>]

gsqcmd help [chm]
```

Формат параметра <подключение>:

<Имя строки подключения>	Имя подключения из файла gsqcmd.exe.config
or <Строка подключения>	Строка подключения для OLEDB, ODBC, DSN
or <Провайдер>;<Строка подключения>	Строка подключения для провайдеров .NET

Параметры выполнения запросов:

```
/commandTimeout=<количество секунд>
/connectionTimeout=<количество секунд>
/noTransaction
/inputCodepage=<кодировка страницы>
/outputCodepage=<кодировка страницы>
/set=<параметр>=<значение | функция>[;...]
/trace
```

Функции параметров опций /set и /get:

```
UtcNow() | UtcDateTime()
UtcDate()
UtcTime()
Now() | DateTime()
Date()
Time()
NyseDateTime()
NyseDate()
NyseTime()
FileDateTime(<файл>)
FileDateTimeUtc(<файл>)
FileDateTimeNyse(<файл>)
FileDate(<файл>)
FileDateUtc(<файл>)
FileDateNyse(<файл>)
FileTime(<файл>)
FileTimeUtc(<файл>)
FileTimeNyse(<файл>)
FileName(<файл>)
FileNameWithoutExtension(<файл>)
FileText(<файл>)
```

Общие параметры вывода:

```
/addRowNum
/append
/asText
/asCsv
/asHtml
/dateTimeFormat=<format>
/noHeaders
```

Параметры входного CSV файла:

```
/inputSeparator=<разделитель>|Tab
```

Параметры вывода в CSV:

```
/add=<заголовок=значение> [<разделитель>...]
```



```
/[output]Separator=<разделитель>|Tab  
/quoteChar=<СИМВОЛ>  
/escapeChar=<СИМВОЛ>
```

Параметры вывода в HTML:

```
/noTemplate  
/placeholder=<имя форматной вставки>  
/template=<файл шаблона HTML>  
/title=<наименование документа>
```

Параметры генерации кода:

```
/table=<таблица | представление | хранимая процедура> | <файл шаблона SQL кода>  
/fmt=<файл формата>  
/insertIdentity  
/insertNulls  
/singleLineSQL  
/groupSize=<количество строк>  
/keys=<поле>[, <поле>[, ...]]  
/mssql | /sqlce | /mysql | /oracle | /db2 | /nuodb | /pgsql | /sqlite
```

Режимы работы командной строки

Execute

Выполняет запросы SQL или файлы SQL скриптов для заданного подключения и выводит результаты запроса.

Режим используется для выполнения запросов, импорта и экспорта данных.

Данный режим является режимом по умолчанию, и ключевое слово execute может быть опущено.

Prepare

Формирует готовый к исполнению запрос на основе запроса SQL или файла SQL скрипта для заданного подключения.

Далее готовый файл можно выполнить в режиме Execute или с помощью другого инструмента.

Parse

Выводит данные SQL запроса или файла SQL скрипта по командам для целей отладки.

Insert, Update, Delete

Генерирует команды INSERT, UPDATE, DELETE по данным из CSV файла.

Параметр /table позволяет задать целевые таблицу, представление, хранимую процедуру, или файл SQL шаблона, а параметр /fmt задать файл формата с актуальными именами колонок в CSV файле.

Далее, вы можете выполнить сгенерированный код в режиме execute.

Бесплатная версия позволяет генерировать команды для файлов не более 500 строк.

Merge

Генерирует команды INSERT и MERGE для вставки и обновления данных из CSV файла.

Параметр /table позволяет задать целевую таблицу или представление, а параметр /fmt задать файл формата с актуальными именами колонок в CSV файле.

Далее, вы можете выполнить сгенерированный код в режиме execute.

Бесплатная версия позволяет генерировать команды для файлов не более 500 строк.

Fmt+

Генерирует весь набор файлов, необходимый для импорта CSV данных с использованием различных методов.

Этот режим фактически включает режимы: Fmt, Create, Ace, Buck, Ini.

Файл формата создается, если он не существует. Если файл формата существует, то он используется как источник наименований колонок.

Вы можете запустить gsqlcmd в этом режиме дважды: первый раз для генерации файлов как есть, и второй раз после редактирования файла формата.

Fmt

Генерирует файл формата для импорта данных CSV в Microsoft SQL Server с использованием функции OPENROWSET(BULK...) или утилиты bcp.

Файл формата используется в других режимах как источник наименований колонок.

Create

Генерирует SQL код создания таблицы CREATE TABLE для вставки данных из файла CSV.

Параметр /table позволяет задать целевую таблицу, а параметр /fmt задать файл формата с актуальными именами колонок в CSV файле.

Ace

Генерирует SQL код OPENROWSET для импорта CSV данных с использованием провайдера MICROSOFT.ACE.OLEDB.12.0.

Параметр /table позволяет задать целевую таблицу, а параметр /fmt задать файл формата с актуальными именами колонок в CSV файле.

OPENROWSET с использованием MICROSOFT.ACE.OLEDB.12.0 позволяет импортировать данные в кодировке UTF-8, которая не поддерживается командами OPENROWSET BULK.

Bulk

Генерирует SQL код использования функции OPENROWSET(BULK...) для импорта данных CSV.

Параметр /table позволяет задать целевую таблицу, а параметр /fmt задать файл формата с актуальными именами колонок в CSV файле.

Ini

Генерирует описание полей для файла schema.ini, который используется для импорта данных CSV с использованием драйверов ODBC в Microsoft Excel и базы данных.

Параметр /fmt позволяет задать файл формата с актуальными именами колонок в CSV файле.

LocalConfig | UserConfig | AppConfig

Запускает утилиту [gConnectionManager](#) для редактирования подключений в [файле конфигурации](#):

- LocalConfig - gsqlcmd.exe.config в текущем каталоге.
- UserConfig - gsqlcmd.exe.config в каталоге %LocalAppData%\Gartle\gsqlcmd
- AppConfig - gsqlcmd.exe.config в каталоге утилиты.

Connections

Выводит имена подключений из всех файлов конфигурации.

Stamp

Режим может использоваться в командных файлах для установки значений переменных как:

```
for /F %i in ('gsqlcmd.exe stamp') do set stamp=%i
```

Help

Запускает расширенную справку командной строки, а с параметром chm - файл справки в CHM формате.

Для получения краткого описание режимов запуска просто запустите утилиту без параметров.

Позиционные параметры

<Подключение>

Параметр <Подключение> определяет данные подключения к серверу баз данных.

Возможны три варианта:

1. Имя строки подключения, определенное в файле gsqlcmd.exe.config (или gsqlcmd32.exe.config для gsqlcmd32.exe).
2. Строка подключения без указания провайдера для OLEDB, ODBC и DSN.
3. Строка подключения с указанием провайдера, в том числе для .NET провайдеров.

Первый вариант является наиболее удобным, т.к. позволяет указывать понятное имя подключения, а строки подключения хранить в файле конфигурации.

Например, файл gsqlcmd.exe.config содержит следующие строки:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="AzureDemo" connectionString="Provider=SQLOLEDB;Data Source=.\SQLEXPRESS;Initial Catalog=AzureDemo;Integrated Security=SSPI"
      providerName="System.Data.OleDb" />
    <add name="mssql-rtd" connectionString="Data Source=.;Initial Catalog=RTD;Password=r#td_2014_abc!;User ID=rtd"
      providerName="System.Data.SqlClient" />
    <add name="sqlce-rtd" connectionString="Data Source=%LOCALAPPDATA%\Gartle\RealTimeToDB\rtd.sdf"
      providerName="System.Data.SqlServerCe.4.0" />
    <add name="mysql-rtd" connectionString="Server=localhost;Password=r#td_2014_abc!;User ID=rtd;Database=rtd"
      providerName="MySQL.Data.MySqlClient" />
    <add name="ora-rtd" connectionString="Provider=OraOLEDB.Oracle;Password=r#td_2014_abc!;User ID=RTD;Data Source=Oracle/Orcl;PLSQLSet=True"
      providerName="System.Data.OleDb" />
    <add name="db2-rtd" connectionString="Driver=IBM DB2 ODBC DRIVER;Hostname=DB2;Port=50000;Protocol=TCPIP;Database=SAMPLE;Pwd=r#td_2014_abc!;UID=RTD;LONGDATACompat=1"
      providerName="System.Data.Odbc" />
    <add name="nuodb-rtd" connectionString="Server=localhost;Password=r#td_2014_abc!;User=RTD;Database=rtd"
      providerName="NuoDb.Data.Client" />
    <add name="pgsql-rtd" connectionString="Server=localhost;Password=r#td_2014_abc!;User ID=rtd;Database=rtd"
      providerName="Npgsql" />
  </connectionStrings>
</configuration>
```

При выполнении команд могут использоваться имена AzureDemo, mssql-rtd, sqlce-rtd и т.д.

Для редактирования файла конфигурации можно использовать, например, текстовый редактор notepad.exe.

Также можно использовать программный продукт [gConnectionManager](#), который позволяет создавать строки, используя диалог подключения к серверу баз данных.

Для запуска можно воспользоваться режимами localconfig, userconfig, appconfig.

Во втором варианте можно указать строку подключения, если используется OLE DB провайдер, ODBC драйвер или DSN файл.

Например:

```
gsqlcmd "Provider=SQLOLEDB;Data Source=.\SQLEXPRESS;Initial Catalog=AzureDemo;Integrated Security=SSPI" framework-inst
all-en.sql
```

Строка подключения может передаваться в переменной среды. Например, в командном файле можно указать:

```
set connection=Provider=SQLOLEDB;Data Source=.\SQLEXPRESS;Initial Catalog=AzureDemo;Integrated Security=SSPI
gsqlcmd "%connection%" framework-install-en.sql
```

Обратите внимание на кавычки вокруг переменной среды, которые необходимы для экранирования значения одного параметра.

В третьем варианте формата параметра Connection в строке подключения первым указывается имя провайдера, а затем строка подключения.

Это необходимо для определения .NET провайдеров, т.к. их нельзя выделить из строки подключения, как в случаях OLE DB и ODBC.

Пример:

```
gsqlcmd "MySQL.Data.MySqlClient;Server=localhost;Password=r#td_2014_abc!;User ID=rtd;Database=rtd" framework-install-e
n.sql
```

<SQL-запрос> | <SQL-файл>

Вторым обязательным позиционным параметром режима execute является команда SQL запроса или файл скрипта.

Примеры:

```
gsqcmd rtd-ora "SELECT * FROM SYS.ALL_USERS"
```

```
gsqcmd rtd-ora rtd-setup-ora.sql
```

Утилита проверяет наличие файла по имени параметра, и если файл существует, то считывает скрипт из файла.

Это удобно, но может приводить к ошибкам, если указано неверное имя файла, которое отправляется как запрос серверу.

<CSV-файл>

Параметр задает имя входного файла CSV данных для режимов генерации файлов конфигурации и SQL кода.

<Выходной файл>

В позиционном параметре можно задать выходной файл.

По умолчанию, результаты выводятся на консоль.

Расширение выходного файла также используется для автоматического определения выходного формата:

- txt - определяет текстовый формат.
- csv - определяет формат CSV.
- htm или html определяет HTML формат.

Параметры выполнения запросов

/CommandTimeout=<количество секунд>

Задаёт время ожидания выполнения команд сервером.

/ConnectionTimeout=<количество секунд>

Задаёт время ожидания соединения с сервером.

/NoTransaction

Отключает режим использования транзакций при выполнении команд и скриптов.

Транзакционный режим можно отключить, например, при выполнении скрипта удаления объектов БД. Т.к. если некоторые объекты не существуют, это приводит к ошибке выполнения и откату результатов скрипта в целом.

/InputCodePage=<кодировка страницы>

Задаёт кодировку страницы входного файла.

Пример:

```
/inputcodepage=65001
```

/OutputCodePage=<кодировка страницы>

Задаёт кодировку страницы выходного файла.

Пример:

```
/outputcodepage=1250
```

/Set=<параметр>=<значение | функция>[:...]

Параметр позволяет задать значения переменных в командах и скриптах.

Например:

```
gsqcmd mssql-rtd "SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = @Table" /set=Table=RealTimeTables  
gsqcmd mysql-rtd "SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = :Table" /set=Table=real_time_tables
```

Для Microsoft SQL Server, Microsoft SQL Server Compact и SQLite используйте формат @Переменная, для MySQL, MariaDB, Oracle Database, IBM DB2, NuoDB и PostgreSQL используйте :Переменная.

Переменные не стоит использовать со скриптами, которые содержат объявления переменных, которые необходимо передать серверу, например в командах создания хранимых процедур и триггеров.

Тип значения определяется по самому значению.

Если требуется использовать число как строку, используйте одинарные кавычки. Например, '123' вместо 123.

Функции параметров опций /set и /add

Функция	Значение
UtcNow()	Текущие дата и время UTC
UtcDateTime()	Текущие дата и время UTC
UtcDate()	Текущая дата UTC
UtcTime()	Текущее время UTC
Now()	Текущие дата и время
DateTime()	Текущие дата и время
Date()	Текущая дата
Time()	Текущее время
NyseDateTime()	Дата и время торгов на NYSE
NyseDate()	Дата торгов на NYSE
NyseTime()	Время торгов на NYSE
FileDateTime(<File>)	Дата и время файла
FileDateTimeUtc(<File>)	Дата и время файла UTC
FileDate(<File>)	Дата файла
FileDateUtc(<File>)	Дата файла UTC
FileTime(<File>)	Время файла
FileTimeUtc(<File>)	Время файла UTC
FileName(<File>)	Имя файла, включая расширение
FileNameWithoutExtension(<File>)	Имя файла без расширения
FileText(<File>)	Текст из файла

/Trace

Включает режим трассировки, при котором выводятся данные обо всех командах, отправляемых серверу.

Данный режим удобно использовать для понимания конкретной команды скрипта, приводящей к ошибке, или времени выполнения отдельных команд скрипта.

По умолчанию, данные трассировки выводятся на консоль.

Вывод данных можно переопределить в файле `gsqcmd.exe.config` в каталоге утилиты.

Образец вывода в лог-файл приведен в самом файле конфигурации.

Общие выходные параметры

/AddRowNum

Если задан этот параметр, то в данные добавляется первая колонка с номерами строк.

/Append

Если задан этот параметр, то данные добавляются к выходному файлу, а не перезаписывают его.

/AsText | AsCSV | AsHTML

Используйте данный параметр, если формат вывода нельзя определить по расширению выходного файла.

/DateTimeFormat=<формат даты>

Параметр позволяет указать формат для данных типа datetime для вывода в CSV.

См. [http://msdn.microsoft.com/en-us/library/zdtaw1bw\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/zdtaw1bw(v=vs.100).aspx)

Пример:

```
gsqcmd rtd-db2 "SELECT * FROM RTD.QUOTES_YAHOO" QUOTES_YAHOO.CSV /datetimeformat=yyyy-MM-dd
```

Используйте двойные кавычки для указания форматов, содержащих пробелы. Например:

```
"/datetimeformat=yyyy-MM-dd hh:mm:ss"
```

/NoHeaders

Параметр подавляет вывод заголовков таблиц.

Параметры CSV

/Add=<заголовок>=<значение | функция>[<разделитель>...]

Параметр используется для указания дополнительных постоянных данных в выводе CSV.

Пример:

```
gsqcmd rtd-sqlce "SELECT * FROM QuotesYahoo" QuotesYahoo.csv /add=File=QuotesYahoo.csv
```

В примере будет добавлена колонка File с постоянным значением QuotesYahoo.csv.

Функции параметров опций /set и /add

Функция	Значение
UtcNow()	Текущие дата и время UTC
UtcDateTime()	Текущие дата и время UTC
UtcDate()	Текущие дата UTC
UtcTime()	Текущее время UTC
Now()	Текущие дата и время
DateTime()	Текущие дата и время
Date()	Текущая дата
Time()	Текущее время
FileDateTime(<File>)	Дата и время файла
FileDate(<File>)	Дата файла
FileTime(<File>)	Время файла
FileName(<File>)	Имя файла, включая расширение
FileNameWithoutExtension(<File>)	Имя файла без расширения
FileText(<File>)	Текст из файла

/[output]Separator=<разделитель>|tab

Задаёт разделитель данных выходного CSV файла.

Разделитель по умолчанию - точка с запятой.

Используйте значение **Tab** для указания символа tab.

Пример:

```
gsqcmd rtd-mysql "SELECT * FROM rtd.quotes_yahoo" quotes_yahoo.csv /separator=,
```

/inputSeparator=<разделитель>|tab

Задаёт разделитель данных входного CSV файла.

Разделитель по умолчанию определяется автоматически из набора: двойная запятая, запятая, точка с запятой, символ tab.

/QuoteChar=<символ>

Задаёт символ для обрамления строковых значений.

Значение по умолчанию - двойная кавычка.

Обрамление можно отключить, задав /QuoteChar=

/EscapeChar=<символ>

Задаёт символ экранирования символов обрамления строк внутри строковых значений.

Значение по умолчанию - двойная кавычка.

Экранирование можно отключить, задав /EscapeChar=

Параметры вывода HTML

/NoTemplate

Отключает использование шаблона при выводе в HTML.
Фактически, генерируется только таблица выходных данных.

/Placeholder=<имя форматной вставки>

Задаёт имя форматной вставки для вставки выходных данных.

Например, шаблон HTML может содержать вставку {sales}. Вы можете задать опцию: /placeholder={Sales}

Значение по умолчанию: {table}

/Template=<файл шаблона HTML>

Переопределяет шаблон вывода данных в HTML.

Шаблон может содержать форматные вставки, вместо которых производится вставка сгенерированных данных, и которые задаются в параметре /placeholder.

По умолчанию, таблица данных вставляется перед тегом </body>.

/Title=<наименование документа>

Задаёт значение форматной вставки {title} в шаблоне вывода HTML.

Параметры генерации кода

/Table=<таблица | представление | хранимая процедура> | <файл шаблона SQL кода>

Задаёт целевую таблицу, представление, хранимую процедуру или файл шаблона для генерации кода.

Пример: /table=dbo.Payments

/fmt=<файл формата>

Задаёт файл формата для генерации кода.

Файл используется как источник имен полей вместо имен в файле CSV.

Значение по умолчанию - имя файла CSV с расширением *.fmt.

/InsertIdentity

Задаёт опцию генерации вставки полей identity для команд INSERT и MERGE.

/InsertNulls

Задаёт опцию вставки значений NULL в команды INSERT.

/SingleLineSQL

Задаёт опцию генерации однострочных SQL команд INSERT и MERGE вместо групп.

/GroupSize=<количество строк>

Задаёт количество строк в группах между разделителем GO.

/keys=<поле>[,<поле>[,...]]

Опция позволяет указать поля, которые следует использовать вместо полей первичных ключей при генерации команд INSERT, UPDATE, DELETE.

Это удобно использовать при слиянии данных в разных базах данных по суррогатным ключам, типа email или ИНН, вместо числовых первичных ключей.

[/mssql](#) | [sqlce](#) | [mysql](#) | [oracle](#) | [db2](#) | [nuodb](#) | [pgsql](#) | [sqlite](#)

Задаёт целевую платформу баз данных для генерации кода.

Опция используется, если режим не требует указания строки подключения.

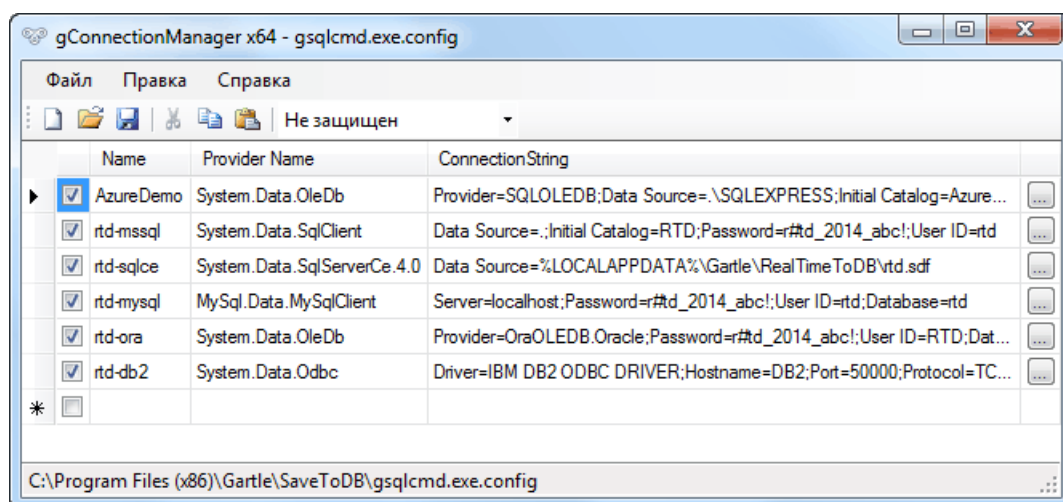
Коды возврата

Код возврата	Описание
0	Успешное завершение
1	Недостаточно параметров для запуска
2	Ошибка в параметрах запуска
3	Исключение в процессе выполнения gsqlcmd
4	Исключение сервера баз данных

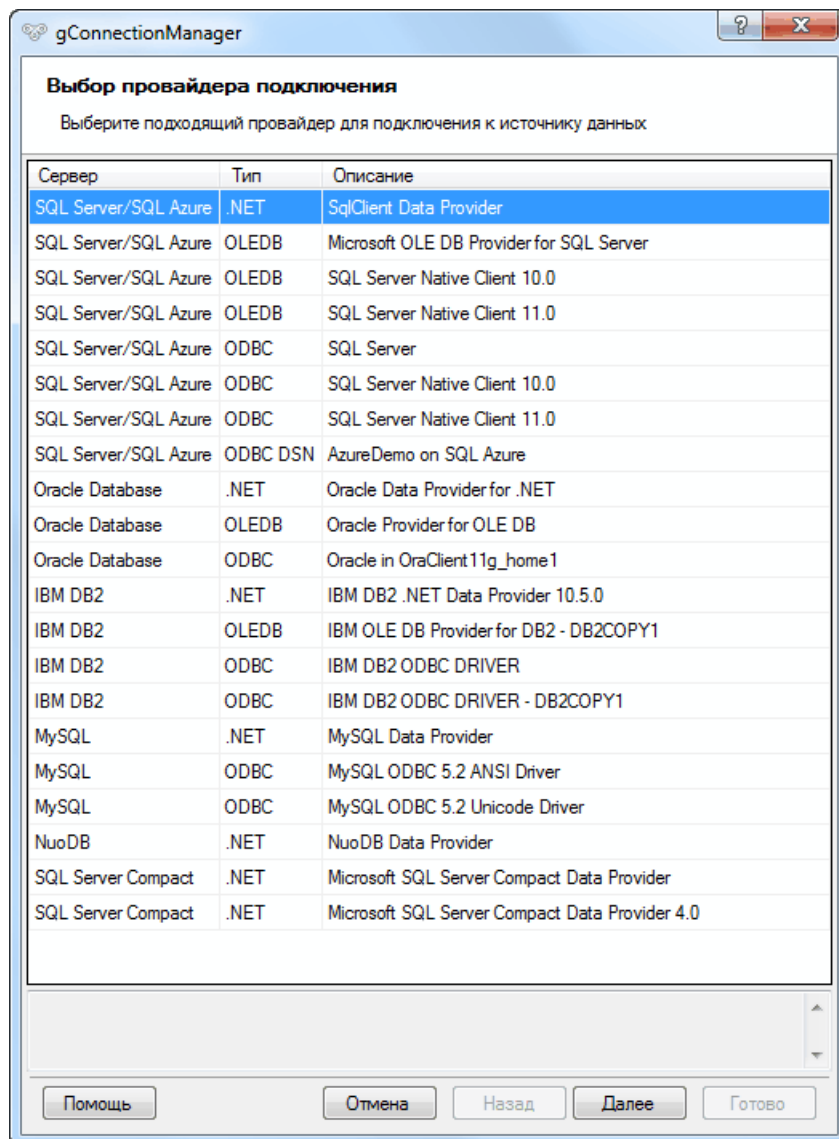
gConnectionManager

Утилита gConnectionManager позволяет редактировать строки подключения в файлах конфигурации в удобном режиме.

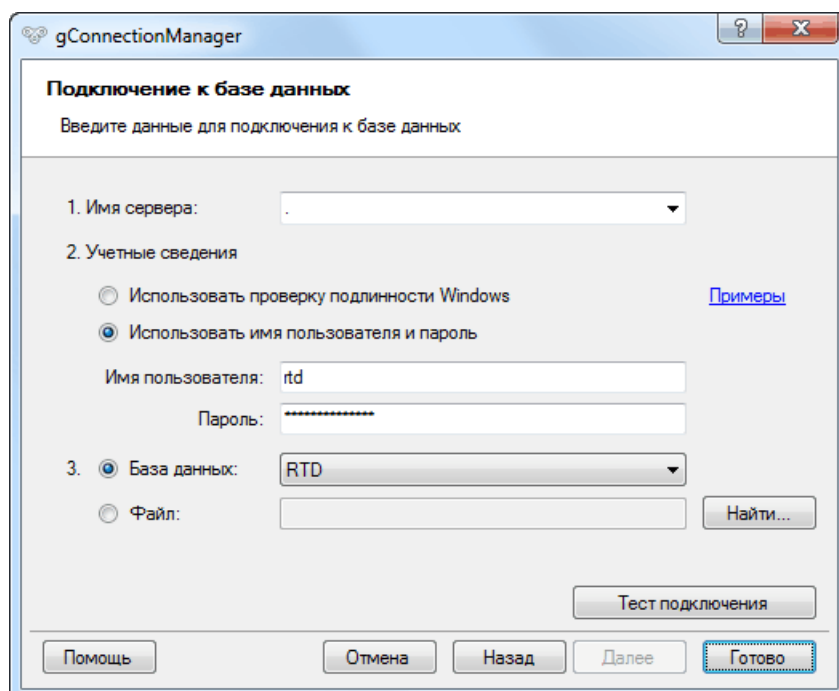
Пример открытого файла конфигурации утилиты gsqlcmd:



При создании или редактировании подключения выбирается провайдер подключения:



На следующем шаге выполняется подключение к базе данных:



Файл конфигурации

Файл конфигурации `gsqcmd.exe.config` позволяет задать строки подключения к серверам баз для дальнейшего использования имен подключений при выполнении команд.

Например:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <connectionStrings>
    <add name="AzureDemo" connectionString="Provider=SQLOLEDB;Data Source=.\SQLEXPRESS;Initial Catalog=AzureDemo;Integrated Security=SSPI"
      providerName="System.Data.OleDb" />
    <add name="mssql-rtd" connectionString="Data Source=.;Initial Catalog=RTD;Password=r#td_2014_abc!;User ID=rtd"
      providerName="System.Data.SqlClient" />
    <add name="sqlce-rtd" connectionString="Data Source=%LOCALAPPDATA%\Gartle\RealTimeToDB\rtd.sdf"
      providerName="System.Data.SqlServerCe.4.0" />
    <add name="mysql-rtd" connectionString="Server=localhost;Password=r#td_2014_abc!;User ID=rtd;Database=rtd"
      providerName="MySQL.Data.MySqlClient" />
    <add name="ora-rtd" connectionString="Provider=OraOLEDB.Oracle;Password=r#td_2014_abc!;User ID=RTD;Data Source=Oracle/Orcl;PLSQLSet=True"
      providerName="System.Data.OleDb" />
    <add name="db2-rtd" connectionString="Driver=IBM DB2 ODBC DRIVER;Hostname=DB2;Port=50000;Protocol=TCPIP;Database=SAMPLE;Pwd=r#td_2014_abc!;UID=RTD;LONGDATACOMPAT=1"
      providerName="System.Data.Odbc" />
    <add name="nuodb-rtd" connectionString="Server=localhost;Password=r#td_2014_abc!;User=RTD;Database=rtd"
      providerName="NuoDb.Data.Client" />
    <add name="pgsql-rtd" connectionString="Server=localhost;Password=r#td_2014_abc!;User ID=rtd;Database=rtd"
      providerName="Npgsql" />
  </connectionStrings>
</configuration>
```

В примере при выполнении команд могут использоваться имена подключений `AzureDemo`, `mssql-rtd`, `sqlce-rtd` и т.д.

Для редактирования файла конфигурации можно использовать, например, текстовый редактор `notepad.exe`.

Также можно использовать программный продукт [gConnectionManager](#), который позволяет создавать строки, используя диалог подключения к серверу баз данных.

В файле конфигурации также можно задать данные провайдера подключения к серверу баз данных, если компоненты провайдера находятся в каталоге утилиты.

Например, в состав `gsqcmd` включен провайдер подключения к MySQL, что настроено в разделе `DbProviderFactories`.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  ...
  <system.data>
    <DbProviderFactories>
      <remove invariant="MySQL.Data.MySqlClient"/>
      <add name="MySQL Data Provider"
        invariant="MySQL.Data.MySqlClient"
        description=".Net Framework Data Provider for MySQL"
        type="MySQL.Data.MySqlClient.MySqlClientFactory, MySQL.Data, Version=6.8.3.0, Culture=neutral, PublicKeyToken=c5687fc88969c44d" />
    </DbProviderFactories>
  </system.data>
</configuration>
```

Регистрация продукта

gsqcmd включает редакции Personal и Enterprise. См. [Сравнение редакций](#).

Если gsqlcmd установлена простым копированием, то для регистрации следует запустить **RegisterProduct.exe** из каталога gsqlcmd.

Если gsqlcmd установлена в интегрированном режиме, то запустите ярлык **Регистрация продукта** в соответствующей группе **gsqcmd** меню Пуск.

Группа gsqlcmd может находиться в следующих папках в зависимости от продукта установки:

- Gartle\gsqcmd
- Gartle\SaveToDB\gsqcmd
- Gartle\RealTimeToDB\gsqcmd

Для регистрации редакций Personal и Enterprise требуется код регистрации, который высылается по электронной почте после приобретения.

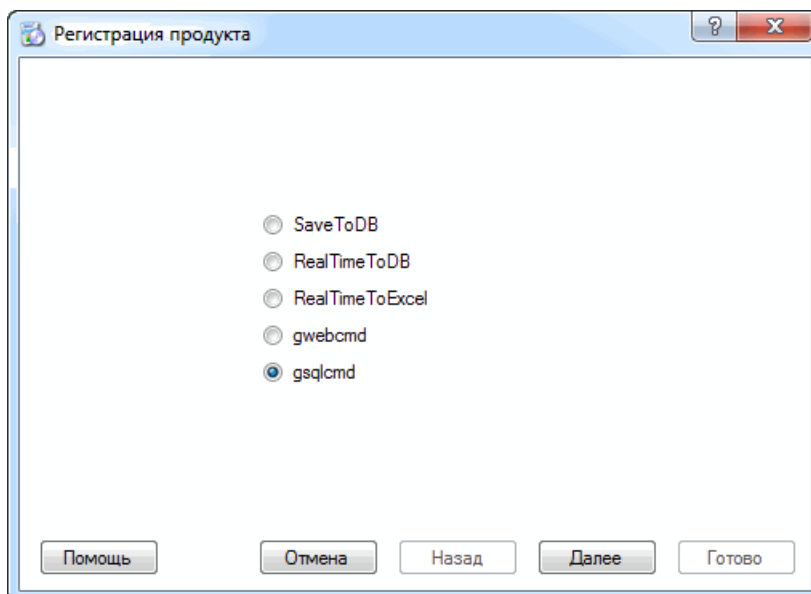
Регистрация продукта в целом включает следующие шаги:

1. Выбор продукта и редакции.
2. Ввод данных владельца лицензии.
3. Активация продукта.

Выбор продукта

Продукты ЗАО "Гартл" имеют унифицированную процедуру регистрации.

На первом шаге выберите продукт для регистрации.



Выбор редакции

Для регистрации редакции gsqlcmd Personal или Enterprise требуется код регистрации. Код регистрации высылается по электронной почте после приобретения лицензии.

Регистрация продукта

Russian

☒ gsqlcmd Personal
☐ gsqlcmd Enterprise

- - -

[Сравнение редакций](#)
[Приобрести gsqlcmd](#)

Помощь Отмена Назад Далее Готово

Ввод данных владельца лицензии

На этом шаге следует ввести данные владельца лицензии и выбрать вариант регистрации.

Регистрация gsqlcmd

Важно: на указанный адрес электронной почты будет выслан код активации.

Имя, отчество: *
Фамилия: *
Эл. почта: *
Компания: *
Страна/Регион: *
Штат/Область: *

☐ Регистрация по e-mail
☒ Регистрация онлайн

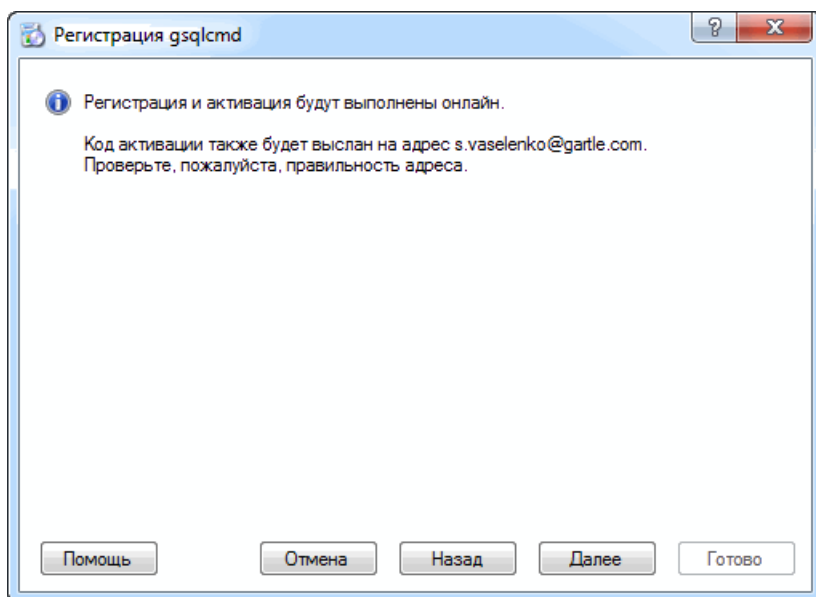
☐ Я хочу получать сообщения о выходе новых версий
☐ Я даю согласие на электронную обработку предоставленных данных
[Политика конфиденциальности](#)

Помощь Отмена Назад Далее Готово

Кнопка "Далее" доступна, если введены все обязательные поля и дано согласие на обработку персональных данных.

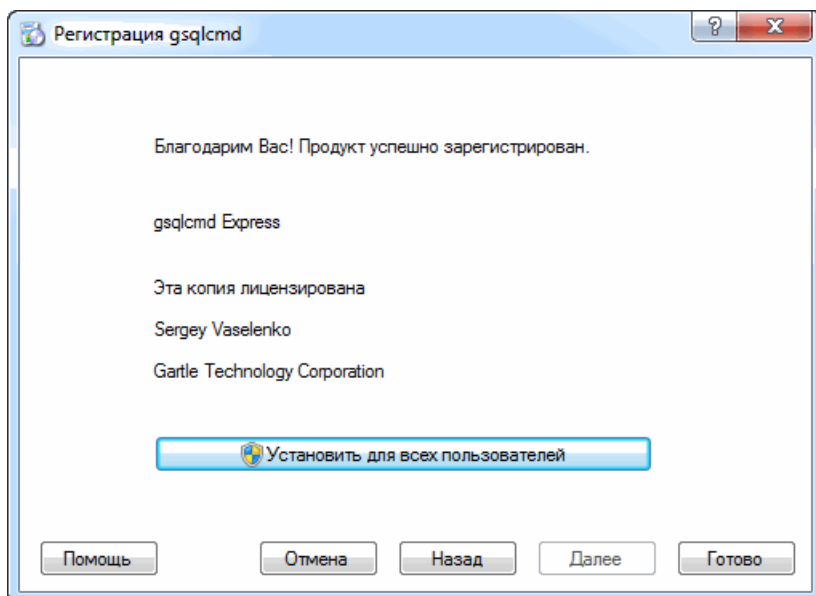
Регистрация онлайн

На этом шаге проверьте, пожалуйста, адрес электронной почты, на который высылается письмо с подтверждением регистрации.



При нажатии кнопки "Готово" плагин gsqlcmd выполняет подключение к серверу регистрации.

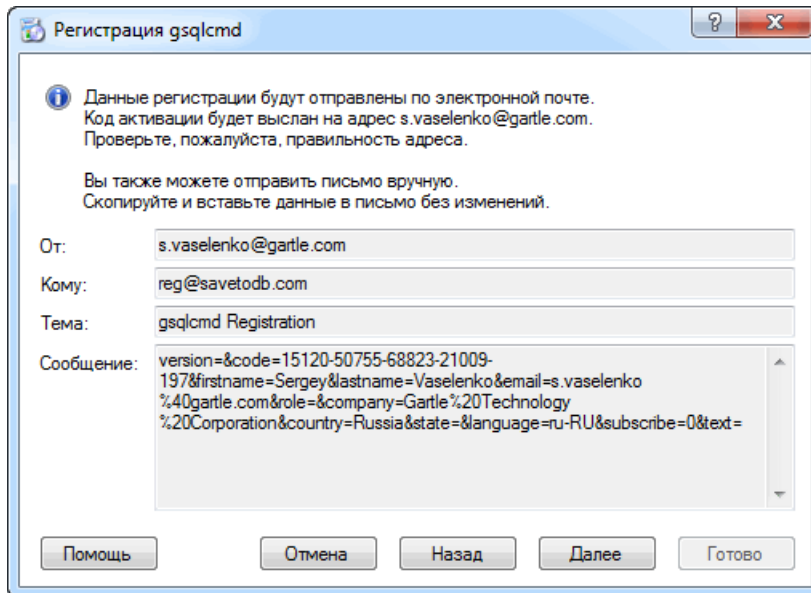
В случае успешной регистрации появляется страница подтверждения успешной регистрации.



Если в процессе онлайн регистрации происходит ошибка, Вы можете попробовать зарегистрировать продукт позже или вернуться на предыдущий шаг и выбрать вариант регистрации по электронной почте.

Регистрация по электронной почте

При выборе варианта регистрации по электронной почте появляется следующая страница:



Регистрация gsqlcmd

Данные регистрации будут отправлены по электронной почте. Код активации будет выслан на адрес s.vaselenko@gartle.com. Проверьте, пожалуйста, правильность адреса.

Вы также можете отправить письмо вручную. Скопируйте и вставьте данные в письмо без изменений.

От: s.vaselenko@gartle.com

Кому: reg@savetodb.com

Тема: gsqlcmd Registration

Сообщение: version=&code=15120-50755-68823-21009-197&firstname=Sergey&lastname=Vaselenko&email=s.vaselenko%40gartle.com&role=&company=Gartle%20Technology%20Corporation&country=Russia&state=&language=ru-RU&subscribe=0&text=

Помощь Отмена Назад Далее Готово

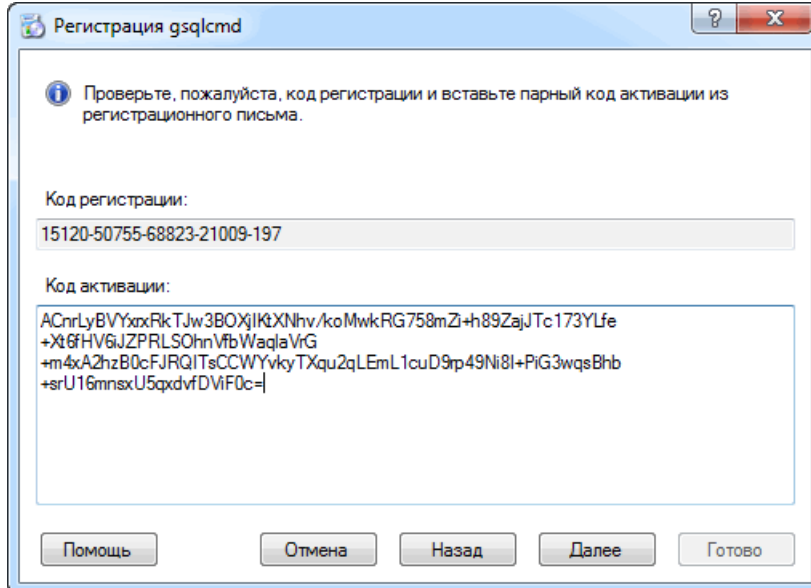
При нажатии на кнопку "Далее" плагин запускает почтовую программу по умолчанию, создает письмо регистрации и переходит на следующий шаг.

Не забудьте отослать письмо регистрации.

Если запуск почтовой программы по умолчанию не получается, то Вы можете сформировать письмо регистрации вручную. Пожалуйста, скопируйте и вставьте в новое письмо поля **Кому**, **Тема** и **Сообщение** без изменений.

Ответ сервера регистрации придет в течение нескольких секунд. Но Вы можете закрыть диалог регистрации и вернуться к нему позже на этом же шаге.

После получения письма, скопируйте и вставьте без изменений код активации в поле "Код активации".



Регистрация gsqlcmd

Проверьте, пожалуйста, код регистрации и вставьте парный код активации из регистрационного письма.

Код регистрации: 15120-50755-68823-21009-197

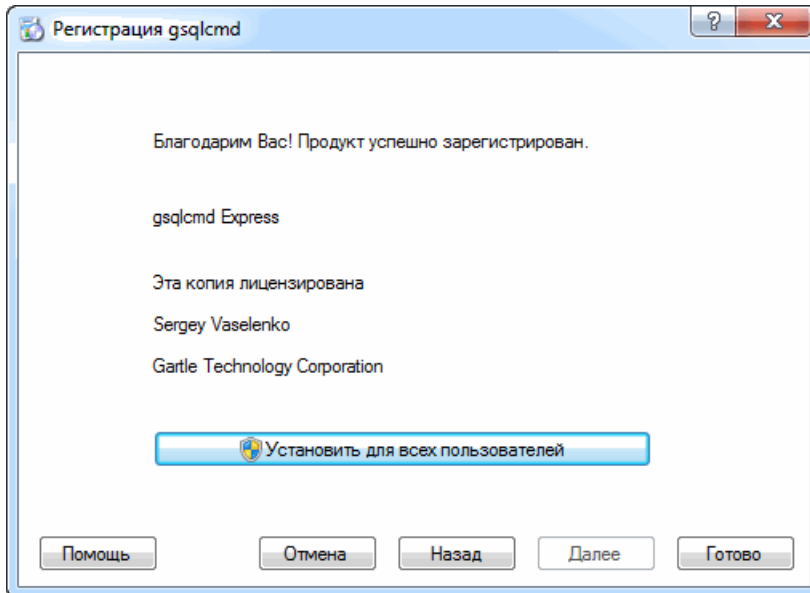
Код активации: ACnrLyBVYxxRkTJw3BOxjIKtXNhv/koMwkRG758mZi+h89ZajJTc173YLfe+Xt6fHV6JZPRLSOhnVfbWaqiaVrG+m4xA2hzB0cFJRQITsCCWYvkyTXqu2qLEmL1cuD9rp49Ni8l+PiG3wqsBhb+srU16mnsxU5qxdvfDVif0c=

Помощь Отмена Назад Далее Готово

Кнопка "Далее" доступна, если код вставлен без ошибок.

Нажмите кнопку "Далее" для завершения процесса регистрации.

Мастер регистрации проверит совпадение данных регистрации и кода активации, и подтвердит успешную регистрацию:



Это все. Нажмите кнопку "Готово".

Техническая поддержка

Новые версии программы можно загрузить с сайта www.gsqlcmd.com.

Вы можете написать нам по электронной почте support@gsqlcmd.com.

Ответы на наиболее часто задаваемые вопросы приведены в разделе [Вопросы и ответы](#).

Вопросы и ответы

Почему поддерживаются только заданные сервера баз данных, если указывается полная строка подключения и команда SQL?

gsqlcmd позволяет выполнять SQL-скрипты и поддерживает возможности, специфичные для поддерживаемых серверов баз данных.

Почему выбран формат указания полной строки подключения, а не по компонентам (сервер, БД и т.д.)?

Строку подключения удобно передавать в переменных среды и можно использовать стандартные возможности работы со строками подключения .NET Framework.

В дополнение, для настройки подключений можно использовать наш продукт gConnectionManager, в котором можно задать сервер, базу данных, имя пользователя и т.д. для создания строки подключения.

Зачем нужны две версии исполняемых файлов gsqlcmd.exe и gsqlcmd32.exe?

В 64-разрядной версии Windows могут быть установлены только 32-разрядные провайдеры для подключения к определенному серверу баз данных.

В этом случае следует использовать gsqlcmd32.exe. Данные подключения в этом случае задаются в файле gsqlcmd32.exe.config.

При выполнении скрипта выводится сообщение об ошибке. Как отладить скрипт?

gsqlcmd позволяет выполнять скрипты на уровне утилит от производителей баз данных для стандартных команд DDL и DML, включая наиболее важные расширения синтаксиса.

Если скрипт не работает, проверьте его при работе через утилиту от производителя или в среде разработки.

Воспользуйтесь опцией /trace для получения точного текста команды SQL, отсылаемой серверу.

Если Вы считаете, что не работает штатная возможность, вышлите, пожалуйста, нам пример такого скрипта.