

GetMsSqlDump Documentation

Version 0.3

Changelog:

Date	Version	Comment
2009-01-22	V0.1	First version of the tool and the document
2009-03-31	V0.3	Updated with changes (see release notes at the end)

Quick Reference

GetMsSqlDump v0.3 - A data dumping tool for Microsoft SQL Servers,
by <Bitemo, Erik Gergely>, 2009 <http://blog.rollback.hu>

Usage:

```
(powershell) GetMsSqlDump.ps1 [-server servername] [-db dbname]
               -table tablename [-query "customquery"] [-username username -password password]
               [-file filename] [-dateformat dateformat]
               [-append] [-noidentity] [-debug] [-help] [-?]
```

Description

This is a tool which enables you to dump the content of one or more tables into a text file in the form of INSERT INTO statements, allowing you to archive/transfer/review/modify the data in an easy and convenient way.

Parameters

servername : Name of database server to connect, port other than 1433 should be added with a comma (e.g. SQL01,1435). At the moment, protocols cannot be specified.

Default value: localhost.

dbname : Name of the database to connect to. If missing, the user's default database will be used.

No default value.

tablename: Name of table(s) to dump. You can use the * (asterisk) as wildcard which will be translated into the % wildcard during pattern matching. Note that the schema (or owner in pre-SQL 2005 versions) is part of the name. Wildcards work with pre-SQL 2005 versions now. If you want to dump all the tables, just type a *. If you use a custom query, tablename will be the name of the new pseudo-table the insert commands will target.

No default value.

customquery: An arbitrary SQL query which returns one or more result set(s). In case of multiple result sets the first result set will get the name specified by the -table parameter, the subsequent ones will get the specified name suffixed by an underscore and the 0-based ordinal of the result set. That is, if you specified tbl as the table name and you have 3 result sets, they'll be called tbl,

tbl_1 and tbl_2. If you don't specify a tablename, the built-in default is Qry. If you don't specify column names for computed columns, they'll get the name column<ordinal> name where <ordinal> shows the ordinal of the column among the unnamed columns.

No default value.

username: SQL login name if SQL authentication is used. If no value given, Windows integrated authentication will be used and the password parameter will be ignored.

No default value.

password: Password of the SQL login specified in the username parameter. If no username was specified, this parameter will be ignored.

No default value.

file: Destination of the dump file. If omitted, dump will be redirected to stdout. If the file already exists, either the -Append or the -Overwrite switch should be specified. Submitting both switches results in script abortion to avoid ambiguous situations and unintentional data loss.

No default value.

dateformat: Format of datetime fields in tables. For all the options please refer to the MSDN "Custom DateTime Format Strings" on the web. For basic tutorial, go down to the **dateformat options** section.

Default value: "yyyy-MM-dd HH:mm:ss.FF"

Switches

Switches are Boolean parameters without arguments, if they present, their value will be true.

-append	Dump will be appended to the file specified by file parameter.
-overwrite	Dump will overwrite the file specified by file parameter.
-noidentity	Identity values won't be dumped. This way you can add the rows to a table with the same identity column specification. If no identity column exists in the table, the switch will be ignored.
-debug	Prints way more characters to your screen than you'd like to. If something didn't work in the way you expected, or you want to submit a bug, run your statement with the debug switch.
-help	Prints this short help. Ignores all other parameters.
-?	Just like help, as long as it is the only parameter.

Examples

Example 1: **GetMsSqlDump.ps1 -server SQL01 -db AdventureWorks -table Person.Address -file C:\Documents\Address.sql -Overwrite -Noidentity**

This command will dump the content of the Person.Address table (omitting the identity column) from the AdventureWorks database on server SQL01 and will write it into the file C:\Documents\Address.sql. If the file already exists, it will be overwritten and all of its content will be lost.

Example 1b: **GetMsSqlDump.ps1 -s SQL01 -d AdventureWorks -t Person.Address -f C:\Documents\Address.sql -o -n**

It does the very same thing, but a bit shorter. In PowerShell, you must specify just enough characters from the parameter name to make it unambiguous for the shell. You can even omit the parameter names if you specified all the parameters in the expected order.

Example 2: **GetMsSqlDump.ps1 -server SQL01 -db AdventureWorks -table Person.* -file C:\Documents\PersonSchema.sql -Overwrite -Noidentity**

This will dump all the tables under the Person schema in SQL 2005 and above and will dump the tables owned by the Person user in SQL2000 and below.

Example 3: **GetMsSqlDump.ps1 -server SQL01 -db AdventureWorks -table CustomTable -query "select top 100 contactID, FirstName, MiddleName, LastName from Person.Contact" -file C:\Documents\PersonSchema.sql -Overwrite -Noidentity**

This will run the specified query and will save its result as insert commands into the CustomTable table

Additional information

Default parameters

All the parameter defaults can be set at the very beginning of the script.

Dateformat options

The dateformat string can be built from strings specifying the formatting of individual dateparts. The string is case sensitive, for example m is for minute and M is for Month.

Year: y

y|yy|yyy|yyyy Specifies the year in as many digits as the number of y characters. In 2009, the y will return 9 and yyy gives 009. You can use yyyyy as well.

Month: M

M|MM Returns the month as a number. MM adds a leading zero if the month is less than October (that is, its ordinal is less than 10).

MMM Returns the abbreviated name of the month.

MMMM Returns the full name of the month.

Day:

d Returns the day of the month as a number between 1 and 31.

dd Returns the day of the month as a number between 1 and 31. If the number is less than 10, a leading zero will be added.

ddd Returns the short name of the day of the week.

dddd Returns the full name of the day of the week.

Hour:

h | hh Returns the hour as a number between 1 and 12. The string hh adds a leading zero if the hour is less than 10.

H | HH Returns the hour as a number between 0 and 23. The string HH adds a leading zero if the hour is less than 10.

t | tt Displays one/two characters of the AM/PM designator.

Minute:

m mm	Returns the minute as a number between 0 and 59. The string mm adds a leading zero if the minute is less than 10.
Second:	
s ss	Returns the second as a number between 0 and 59. The string ss adds a leading zero if the second is less than 10.
f ff fff ffff	Returns the fragment seconds with the given precision. Trailing zeros will be added.
F FF FFF FFFF	Returns the fragment seconds with the given precision. Trailing zeros won't be added.
z zz zzz	Displays timezone info. z and zz show hours only, zzz shows hours:minutes.

Known issues and improvement areas

Now this is a nice list ☺

1. No error handling at the moment – you may end up in not-so-informative PowerShell/.NET exceptions.
2. ~~Wildcard doesn't work on SQL 2000/7.0.~~ fixed in 0.2
3. You can't use ? as wildcard.
4. You can't set the output file encoding.
5. It works only with PowerShell. There will be a "native" C# version as well.
6. All the output goes into a single file which can be inconvenient in case of 50 tables. There will be an option to split the output into different files based on the source tables.
7. ~~You can't filter out records from a table, it's an all or nothing dump. An option will be added which will enable you to submit any kind of queries, including joins.~~ Fixed in 0.2

Things which are not on the to-do list

1. Scripting out table structure – you can do this by using built-in SQL Server management tools and they really do the job. I'm not going to reinvent the wheel.
2. Database security: this tool doesn't bother with security. Sorry. If you have the permission to directly access the database, you have way better options to hack the database than using this tool.

Contact, Bug Report

If you want to contact me in regards to this thing, you can reach me through my website:

<http://blog.rollback.hu/>. You can find the latest-greatest version of this tool on CodePlex at <http://getmssqldump.codeplex.com>.

In case of a bug submit or unexpected behavior, please run the tool with the –Debug switch.

Release notes:

Version 0.3 (John Paul II):

- Handling of multiple result sets in custom queries
- Table header includes creation date, server and database name
- Documentation updated

Version 0.2 (internal):

- Wildcard in table names works with all versions of SQL Server
- Custom query can be scripted – you can run arbitrarily complex (or as simple as a top 100) T-SQL queries to dump out their result sets

Version 0.1 (St. Paul):

- initial release, still in beta