

# MiniM CommandLine Tools

Eugene Karataev  
mailto:support@minimdb.com  
http://www.minimdb.com

November 20, 2016

## 1 MiniM CommandLine Tools

MiniM CommandLine Tools is a set of console utilities, which call the MiniM Database Server for command executing, export and import routines, and routine compilation. Utilities runs on the client computer, but call data on the server. This set of utilities made for Windows 32-bit, Linux i386, Linux x86-64, Linux ARM 32-bit, Linux ARM 64-bit, FreeBSD i386, FreeBSD x86-64 and MacOSX.

For calling server utilities use special dynamic - link library MiniM Server Connect (minimsc.dll for Windows and libminimsc.so for Linux). As a server may be used any MiniM Database Server instance, on the same or other operating system.

All utilities gets parameters from the command line:

```
Usage: programname [-s server] [-p port] [-d database]
server:   TCP/IP computer name of MiniM Server.
port:    TCP/IP port of the server (default 5000).
database: current MiniM database name.
```

Instead of server name may be used network name of the server as such as TCP/IP address of them. All three parameters to connect server may be omitted, and in this case utilities use connection parameters from the ini file mnmct.ini in the Connection section:

```
[Connection]
Server = localhost
Port = 5000
DataBase = user
```

This file with connection settings (if present) must be in the current directory, where utilities run.

Most of this utilities use parameters as a list of name masks - routines and file names. Masks can use special placeholder symbols:

\* - any symbol sequence  
? - any one symbol

For routine names utilities use special extensions:

INT - intermediate code on the MUMPS language  
MAC - macrocode on the MUMPS with  
preprocessor directives  
INC - includable macroroutines

Command-line parameters must be specified in dependence of the client operating system conventions about special symbols. In particular, in the Windows may be used double quotes

```
mnmdbexec.exe "w $zv,!"
```

and for Linux single quotes

```
./mnmdbexec 'w $zv,!'
```

Quotes are mandatory for Linux version if parameters contains mask placeholders.

All utilities supports special option -h, and if found it, does nothing and displays usage conventions. The same screen displays if utility cannot determine connection parameters from command line or ini file and if cannot connect to the server.

All MiniM CommandLine Tools works without transactional context, so cancelling changes made in the database is impossible.

## 2 Utility **mnmdbcompile**

Utility **mnmdbcompile** (**mnmdbcompile.exe** for Windows) runs and executes the compilation of all routines specified in the argument. All routines are compiled on the server, and compilation result outputs to the display. Also compilation result may be redirected to the file or to other application.

Utility uses argument as a list of names masks, comma-delimited. In the mask may be used placeholder symbols.

All INC routines does not compiles. INT routines translates as routines with pure MUMPS code. MAC routines translates by macroprocessor and next compiles as INT routines. If argument contains two routines with INT and MAC type, utility translates only MAC routine. If argument contains INT and MAC routines, INT routines translates first, and MAC routines translates second. This convention may be need to use ready INT routines in the macrocode.

If one or more routines contain errors, utility does not stop compilation and continues translation and outputs full compilation report.

### 3 Utility mnmdbexec

Utility mnmdbexec (mnmdbexec.exe for Windows) runs and executes one line of MUMPS commands from the argument.

Line with MUMPS commands cannot contain preprocessor directives.

Utility's argument can contain not a MUMPS commands, but indirection symbol (@) and followed file name with series of lines with MUMPS commands. In this case utility executes line from the file sequentially, line by line.

Line with MUMPS commands executes as arguments of the XECUTE command, so execution of the NEW command have visibility only in the current line, not in other lines. Also cannot be used lines with labels and dotted lines. Jumps by GOTO command also are impossible.

#### Example

```
mnmdbexec.exe "w $zv,!"  
MiniM for Windows x64 1.15 release build
```

### 4 Utility mnmdbkill

Utility mnmdbkill (mnmdbkill.exe for Windows) deletes routines from the database on the server. Utility removes only routines with names matched to routine masks specified in parameter.

#### Example

```
mnmdbkill.exe add*.int  
Kill routine addrou.int? [ Yes | No | All | Stop ] : y  
Routine addrou.int removed.  
Routine addrou.obj removed.
```

On removing utility asks action need to be done. List of supported actions:

- Yes Utility kill this routine.
- No Utility don't remove this routine.
- All This and oll next routines matched to masks will be removed without confirmation.
- Stop This routine and all next routines will not be removed and utility stops execution.

If this utility was run in noninteractive mode (for example from the batch file or with input redirection), utility does not ask any confirmation and stops execution.

On removing routine this utility made actions in depend of routine type:

- INC Kill only INC routine.
- INT Kill this INT routine and derived OBJ bytecode.
- MAC Kill this MAC routine and derived INT routine and OBJ bytecode.

## 5 Utility `mnmdblist`

Utility `mnmdblist` (`mnmdblist.exe` for Windows) outputs list of routine names, with names matched to masks in the argument. If routine matches two or more masks, this name outputs only once.

### Example

```
mnmdblist.exe %aN*
%aN*.INT
%aNCA.INT
%aNCB.INT
%aNCC.INT
%aNCG.INT
%aNCG*.INT
%aNCL.INT
%aNCLEng.INT
%aNCLRus.INT
%aNCU.INT
%aNCUF4.INT
%aNCUF5.INT
%aNCUF9.INT
```

`%aNCW.INT`  
`%aNOT.INT`

If routine extension was not specified, utility uses INT routines. Output contains routine extensions anyway.

## 6 Utility `mnmdbrfirst`

Utility `mnmdbrfirst` (`mnmdbrfirst.exe` for Windows) runs and outputs first lines of routines, which names matches to masks in the argument. If name matches two or more masks, this name uses only once.

First lines of routines contains in most cases identification information about routines, which is used by developers or administrators and have short description of the routine purpose. This is not mandatory rule, but most of developers and developer's tools follows them.

Utility outputs routine name and, in the same line, first routine's line.

## 7 Utility `mnmdbri`

Utility `mnmdbri` (`mnmdbri.exe` for Windows) do import routines from files to the database on the server. Utility uses the following rules for files and routines:

1. All file content is entire routine content, without splitting to several routines in one file.
2. Files with extensions correspondes INC routines, `.int` - INT routines, and `.mac` - MAC routines.
3. Timestamp of modification of routine on the server correspondes timestamp of file modification in the file system.
4. Utility looks up routines in the current directory.
5. Routine name correspondes to the file name.
6. The percent symbol at beginning of routine name correspondes to underscore symbol at beginning of file name.

Utility supports parameter `-m mode`, where instead of mode may be used value 'all'. In this case utility do import of all routines with names matched

to masks and ignores file modification timestamp. Otherwise utility compares file modification timestamp and routine modification timestamp and process file only if routine on the server does not exist or file has older modification timestamp than routine.

On import INC routines utility do import only, and on import of INT and MAC routines utility do import and automatic compilation and outputs compilation errors if they exist.

**Example:**

```
mnmdbri.exe *.int
Importing text.int
Compiling text.int
```

## 8 Utilities mnmdbriFMT

Utilities mnmdbrigt and mnmdbrisa (mnmdbrigt.exe and mnmdbrisa.exe for Windows) make import of routines from files to the database.

Utilities get from the argument comma-separated list of file masks, where may be used placeholder symbols.

Utility mnmdbrigt uses special file format, which is used in the GT.M system - only one routine per file and without special leading and trailing lines. File name uses as a routine name. Files must have the .m extension. If first symbol of the routine is a percent, then first symbol of the file must be underscore.

Utility mnmdbrisa uses file format as specified in the MUMPS standard and in the RSA format. First lines must be special lines, file can contain set of routines, and in this case utility imports all routines from the file.

RSA file format differs from the standard file format by additional information about routine type (INT, MAC or INC) and datetime of change of routine.

If utility imports routines from the RSA file, utility sets changing datetime as specified in the file, and if imports from the standard file, changing datetime sets by the file's datetime.

Utility mnmdbrigt sets the routine change datetime by the file change datetime.

If utility mnmdbrisa uses standard file format, utility creates INT routine, as such as mnmdbrigt. This type of routine is the same as specified in the MUMPS standard.

**Example**

```
mnmdbrigtm.exe *.m
Processing file A1B2MUT.m
Importing routine A1B2MUT.INT
Processing file AAQJIDOC.m
Importing routine AAQJIDOC.INT
Processing file _DT.m
Importing routine %DT.INT
Processing file _DTC.m
Importing routine %DTC.INT
Processing file _ZTRDEL.m
Importing routine %ZTRDEL.INT
```

Utilities mnmdbrisa and mnmdbrigtm uses only files on the client computer. If files are placed on other computer, developer must mount this directory as a network drive or a part of a file system in depends of the currently used operating system.

## 9 Utility mnmdbro

Utility mnmdbro (mnmdbro.exe for Windows) do export of routines from database to files in the current directory. Utility uses the same rules rules for files and routines as utility mnmdbri:

1. All file content is entire routine content, without splitting to several routines in one file.
2. Files with extensions correspondes INC routines, .int - INT routines, and .mac - MAC routines.
3. Timestamp of modification of routine on the server corresponds timestamp of file modification in the file system.
4. Utility looks up routines in the current directory.
5. Routine name corresponds to the file name.
6. The percent symbol at beginning of routine name corresponds to underscore symbol at beginning of file name.

Utility supports parameter -m mode, where instead of mode may be used value 'all'. In this case utility do export of all routines with names matched

to masks and ignores file modification timestamp. Otherwise utility compares file modification timestamp and routine modification timestamp and process file only if file in current directory does not exist or routine has older modification timestamp than correspondent file.

## 10 Utilities mnmdbroFMT

Utilities mnmdbrogtm, mnmdbrorsa and mnmdbrostd (or set of utilities mnmdbrogtm.exe, mnmdbrorsa.exe and mnmdbrostd.exe for Windows) do export of routines from the database to the file.

Utilities use to write files on the local computer. If files must be placed on other computer, this directory must be mounted as network file drive or a part of a file system in depends of used operating system.

Utilities mnmdbrorsa and mnmdbrostd gets from the first argument list of routine name masks, and second argument uses as a file name. All routines matched to masks this utilities exports to this single file. Utility mnmdbrogtm accepts only one argument, with routine name masks. File names for export are derived from the routine name. If first symbol of routine is a percent, first symbol of file name will be underscore and extension will be .m.

If developers use utility mnmdbrogtm, they need to understand that the routine names are case-sensitive, but not all file system have this option. Moreover, files with exported routines may be processed by different external tools, which can change case of symbols automatically. If this do, developers must use other file format.

On export routines by the mnmdbrorsa utility save routine type and modification datetime.

On export routines by the mnmdbrostd utility exports only routines with INT type and datetime of changes does not store in the file.

On export routines by the mnmdbrogtm utility saves routine modification datetime in the file's modification datetime.

## 11 Authorization

Utilities on connecting to the MiniM checks that the user authorization is required. By default MiniM Database Server installs without authorization and GUI utilities does not use login and password.

Authorization can be enabled in minim.ini file on the server, if in the [Login] section will be nonempty string used as MUMPS expression. If

this string is not empty string, utilities ask user about login and password and writes to local variables %username and %password. And next utilities evaluates specified expression.

**Example:**

```
[Login]
GUI = $$checkuser^checkuser(%username,%password)
```

If on evaluation result is non-zero value, utilities count it as authorization success and continues to work. Otherwise utilities stops execution.

Authorization expression in minim.ini may be changed without restart of the server.

If authorization was made, this procedure will be used by MiniM Client Tools, such as MiniM Routine Editor, MiniM Global Editor and MiniM Control Center.

## 12 Version Control Systems

Utilities from the MiniM CommandLine Tools may be used to organize administration (tools for administrators) as such as development (tools for developers) on the MUMPS language. Administration and development tools has different purposes:

1. Administrators need to transfer entire routine set by one file, but developers works with separate routines.
2. For administrators is not important that files have additional content which does not write developers, but for developers it is very important for merging changes by version control system.
3. Typical job for administrators is import and recompilation of entire routine set and time of this may be anyone, but developers need to export and import only changed routines, and time of this need to be small.

Considering these requirements, mnmdbroFMT and mnmdbriFMT utilities are oriented to administrators and utilities mnmdbro and mnmdbru are oriented to developers.

Typical administration operation is import or export entire set of routines and full recompilation. And typical developer's operations are changing only small subset of routines (one or several routines), and for this changes need to be applied version control system.

All version control systems works with files in file system and operates with file content, change timestamp and file name. So, need to be implemented routine synchronization with files in file system without additional automatic-added file content.

Typical developer's job scenario can be the following:

1. Developer updates files from version control system's repository in the work directory and execute synchronization with routines in the database on the server using mnmdbri utility.
2. Developer edits, compiles, executes and tests application.
3. Next changed routines need to be exported to file system using mnmdbro utility.
4. For blocking version control systems need to be applied operation 'check in'.
5. For non-blocking version control systems need to be applied 'update' operation with resolving merge conflicts (if they are present) and following 'commit' operation.

Working with version control system allows to developers use different MiniM Database Server instances for each of developers and make changes, compilation and testing routine set independently of each other, with using version control system merging rules.

At the same time, mnmdbroFMT utilities allows to get full installable routine set for administrators.