msql

Christophe Sollet

COLLABORATORS			
	TITLE : msql		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Christophe Sollet	April 15, 2022	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

msql

1.1 msql.library

```
                          Msql.library V5
©1999 Christophe Sollet
  All Rights Reserved.


What~is~mSQL?

What~is~msql.library?

Licence

Requirement

Install

ARexx Support

mUSD

Author

Thanks

Know Bugs

Users of version older than 2.0
web:http://altern.org/cfc/msql
http://altern.org/amsql
```

1.2 What is mSQL?

```
From the msql introduction documentation:

Mini SQL 2.0 is the second generation of the mSQL database system.  The first
```

generation product, mSQL 1.x, was designed to provide high speed access to small data sets. The original goal was to perform 100 basic operations per second on an average UNIX workstation with small data sets using very few system resources (i.e. memory and CPU cycles). The original design goal was met and the software has proven to be popular because of this functionality.

During mSQL's life, people have used it for applications far beyond the scope of the original design. These high-end applications, containing up to 1 million rows of data, showed a need for better handling of complex queries and large data sets if the package was to be used in this way. The second generation of the mSQL server has been designed to suit these high-end applications while maintaining the original design goals of mSQL 1. It has been designed to meet three main criteria

. Provide comparable performance for simple operations as mSQL 1.x.
. Provide rapid access to large databases and complex operations.
. Provide more of the functionality outlined in the ANSI SQL specification.

The introduction from Mike Erasmus:

A SQL Database engine to create database applications. For people who don't know what that means, you can develop example a kick ass financial software system with relations. Without fiddling around in C++ code or lines and lines of code. Just send your query to the engine and it will sort it out for you.

You can example send the following line of code to the "server": SELECT author, title, isbn_no from publishers
 WHERE author = 'Pete Simmons'
 AND title like '%South Africa%'

The above query will then be executed by the server and return all rows which match the WHERE clause, neat eh??

My introduction:

    The mSQL database engine was ported on Amiga by Mike Erasmus. It's the first SQL database engine available on Amiga.
    Ok! Now read the next section!

1.3 What is msql.library?

msql.library is an Amiga shared library that offer the required api to access a mSQL database engine.
    This library can be used by ARexx application too!
    msql.library use any bsdsocket.library available to connect on a distant server. If you have mUSD installed, you can connect to a local server without any TCP/IP stack.
    Without msql.library, you'll need to link your application with the static

msql library (that's mean using gcc and the ixemul.library).
    With it, you can access a mSQL server (running on any platform)
with any language! :)

That's all!

1.4 Requirement

- An Amiga ;)
- OS 3.0 (should work with 2.0, untested)
- a TCP/IP stack (only if you want to connect to a remote server)

1.5 Install

Use the installer script!
(example wasn't copied anywhere...)

Manual install:

User install:
    Copy libs/msql.library libs:

Dev Install (sas/c):
    Copy fd/#? fd:
    Copy include/#? include: all
    Copy doc/#? autodoc:

GCC user:
    Copy the gcc directory content to GG: (assuming you're using GeekGagdet 2)

1.6 Thanks

Thanks go to

    Mike Erasmus (Amiga Port of mSQL, great help about mSQL)
    <horror@smartnet.co.za>
    Check his Amiga mSQL FAQ! (http://members.smartnet.co.za/~horror/mSQL.html)

    Piotr Gapinski (AmigaE version of developer materials)
    <narg@polbox.com>

    And to all others that helped me to improve the library.

1.7 Author

Christophe Sollet

email: cfc@iname.com

web:
http://altern.org/cfc (msql.library support page)
http://altern.org/amsql (Amiga mSQL Internet Ressources)

1.8 Licence

THIS SOFTWARE IS PROVIDED "AS IS". NO REPRESENTATIONS OR WARRANTIES ARE MADE WITH RESPECT TO THE ACCURACY, RELIABILITY, PERFORMANCE, CURRENTNESS, OR OPERATION OF THIS SOFTWARE, AND ALL USE IS AT YOUR OWN RISK !

The msql.library is Copyright ©1999 by Christophe Sollet

- You may use this library for free if your product is free to the user (Freeware, Mailware, Public Domain, etc...)

- If your application is Shareware or Commercial, you should send me the full release or a keyfile.

You are allowed to redistribute this package with the following condition:

- Everything in this distribution must be kept together, in original unmodified form.
- You may include the library file in your own distributions, if you follow the copyright rules stated above.

1.9 Know Bugs

None yet... Please mail-me if you found one!

1.10 Users of version older than 2.0

Since the library doesn't require an external task any more, you can delete the C:msqlci file.

In addition, the MsqlConnection structure was used in previous version to kept information about the connection between the library and the sub-task. So, the MsqlConnection argument of some function was not used anymore and prototypes have been modified (the data manipulation functions in fact). Check autodoc to correct your source code. Of course, this version is fully backward compatible with previous version.
If you want to compile without any change, just add a
#define MSQL_OBSOLETE_PROTOS

before including msql stuff.
The support of this old protos may be removed soon...

1.11 ARexx Support

msql.library provides an ARexx function host interface that enables ARexx programs to access mSQL Database server. The functions provided by the interface are directly related to the library functions.

The function host library vector is located at offset -30 from the library. This is the value you provide to ARexx in the AddLib() function call.

For more informations, read the AutoDoc.

1.12 mUSD

mUSD means mSQL Unix Socks Deamon. It's a gateway between the msql.library and ixemul unix (or "local") socks used by mSQL. So, you can run mSQL and app's that use msql.library without any TCP/IP stack. Last, but not least, communications using mUSD are 3 to 4 times faster than TCP/IP ones.

The program "speed_test" in the example directory selects the first db and makes the same query 100 times on the first table, one time using the TCP/IP stack, one time using mUSD... Just read times result!

To use mUSD, the msql.conf must have a line like that:
Local_Access = True

If you want to run mSQL without tcp/ip stack, the msql.conf must have a line like that:
Remote_Access = False

To run speed_test, mUSD must be running. In addition, Local and Remote Access must be set to True. Of course, a tcp/ip stack must be running.

If you don't run mUSD, the msql.library will use the tcp/ip stack to connect on a local server.

To kill mUSD send it a break signal.

Arexx users: To make a connection using mUSD, just forget the second argument of the MsqlConnect() function.