

# **Migrating by using User Defined Functions**

A large, faint, light gray watermark of the MySQL fish logo is visible on the left side of the slide.

**Jan Kneschke**  
**Consulting, MySQL AB**  
**jan@mysql.com**

**MySQL UC 2005**  
**Santa Clara**

# Overview

- User Defined Functions (UDF)
  - Use of UDFs
  - Writing UDFs
  - Using UDFs
  - Limitations of UDFs

## Who is this guy ?

- Jan Kneschke
  - Lives in Kiel, Germany at the Baltic Sea
- Consultant and Trainer in the Germany and Europe
- In the future he will be part of the Merlin team
- Develops a fast, low-resource and flexible webserver called `lighttpd` (called `lighty`) in his spare time
- Speaks assembler, C, C++, PHP and Perl

## Questionnaire

- Who is using MySQL ... ok, scratch that. :-)
- Who is using User Defined Functions ?
- Who is using MySQL 5.0 or newer ?
- Who is using a older version of MySQL ?
  - 3.23.x
  - 4.0.x
  - 4.1.x
- Which programming language are you using to access MySQL ?
  - Java, PHP, Perl, C#, C/C++, Ruby

## Use of UDFs

- Extending functionality of MySQL function set
- Accessing external libraries
  - Everyone wants to send mail from the database
  - Interacting with external applications in combination with Triggers
  - Adding new aggregational functions

## Migrating from Oracle

- Oracle provides some functions which MySQL doesn't
- You can add them by adding UDFs
- Examples:
  - CORR
  - INITCAP
  - REGR\_
  - VAR\_POP
- Hartmut Holzgraefe ([hartmut@mysql.com](mailto:hartmut@mysql.com)) implemented most of them by using UDF\_Gen

## UDF\_Gen

- UDF\_Gen was ported from PECL\_Gen to MySQL to generate UDFs more easily
- [http://php-groupies.de/UDF\\_Gen/](http://php-groupies.de/UDF_Gen/)
- Takes a XML file containing the structure and parts of the code
- Generates C++-Code, configure script and Makefile
- Prepares Function-prototypes into MySQL

## Writing UDFs

- UDFs are shared libraries loaded by the MySQL server at runtime
- Preparations
  - MySQL has to be compiled dynamically (MySQL-Max package)
  - If you build yourself:
    - `--with-mysqld-ldflags=-rdynamic`
  - A full build environment is required
    - Make, Compiler, Linker
- Thread-Safety
  - Keep in mind that the MySQL Server is threaded
  - Everything has to be thread-safe



## Prototypes

- For a normal stateless function `MAIL()` you have to provide
  - `mail()` your SQL function
  - `mail_init()` called before `mail()` is called the first time
  - `mail_deinit()` called after `mail()` has called
- `mail()` is called for each row, `mail_init()` before the first row, `mail_deinit()` after the last row

```
UPDATE mailqueue
```

```
SET state = MAIL(mailaddr, body),
```

```
last_attempt = NOW()
```

```
WHERE state = 'queued';
```

## mail\_init()

- Is called before `mail()` is called the first time in this query
- Is required since 4.1.10a, optional before
- Allocates necessary memory
- Checks parameters, can prepare type conversion
- Prepares return value and parameters
  - Precision
  - NULL
  - Character Set handling
- Can return an error to specify that an error occurred

## **mail\_deinit()**

- Called after the last row
- Is optional
- Frees all the reserved memory, closes file-descriptors and so on

# mail()

- Is required
- Is doing the real work and is called for each row

```
long long mail(UDF_INIT *initid, UDF_ARGS *args,
               char *is_null, char *error) {
    FILE * fd;
    if (NULL == (fd = popen("/usr/lib/sendmail", ...))) {
        return MAIL_STATE_ERROR;
    }

    /* ... */

    return MAIL_STATE_SENT;
}
```

## Aggregational UDFs

- Used in combination with GROUP BY
- Two additional functions
  - `func_clear()` resets the aggregated value for the group
  - `func_add()` called for each value of the same group

## Adding Functions to MySQL

- Functions are registered once and are reloaded at each start of the server
  - CREATE [AGGREGATE] FUNCTION
  - DROP FUNCTION
- The shared library has to be located in the search path for regular shared libs

```
CREATE FUNCTION mail
```

```
  RETURNS INTEGER SONAME ('mail.so');
```

```
DROP FUNCTION mail;
```

## Limitations

- Can't access MySQL internal structures
  - No tables
  - No databases
- The functions are stateless
- Can be circumvented by using static vars + mutexes
- Don't forget we are threaded

## Summary

- User Defined Functions enable you to extend the functionality of MySQLs function set
- UDFs can access external libraries
- UDFs can't access table-data in MySQL
- UDFs are written in C or C++
- From the users view they behave the same as a ordinary function as `MAX ()` or `SUBSTRING ()`
- `UDF_Gen` simplifies writing UDFs



## Questions

- Any questions left ?
- Feel free to ask me in the lobby
- Or send a mail to [jan@mysql.com](mailto:jan@mysql.com)
- <http://jan.kneschke.de/projects/mysql/>