



USER DOCUMENTATION (ALEPHINO 5.0)

How to automate the Alephino backup procedure



1 Introduction

The fundamental idea of the Alephino backup procedure is to write all database updates (creating, modifying and deleting data) in a log file `../pools/<poolname>.log`. This file always contains the delta between the current database and its last backup.

Such a backup always consists of the identifier file `backup/<poolname>.SID` and backup file `backup/<poolname>.S00`. If a database (also called data pool) exceeds a size of 2GByte, it is automatically split into separate files. Consequently further files named `backup/<poolname>.S01, S02, S03 ... S99` will be created for backup.

Provided that the database and the log file don't become damaged simultaneously (which is quite unlikely), the saving process ensures the **complete restoration** of the data, in case the database itself gets damaged physically or logically, e.g. by a server crash.

Prerequisite for the success of the data recovery is of course the presence of a regular Alephino backup that we recommend to create firstly before you start entering data into a new Alephino data pool.

Our suggestion thus is to regularly shutdown the Alephino-Server (preferably outside the opening hours of the library) and to save the database using the standalone version of the Alephino Administration "adm_batch" while the Alephino server is down and then restart the Alephino server.

The interval for repetition of the backup is hard to estimate since it depends on the quantity of updates in the database which accumulate in the log file as explained above.

This can be automated both under Unix/Linux and under Windows as follows:

1.1 Setting up cron-job for Unix/Linux

In the standard, the distributed version already includes the following files:

- a command file "savepool.cmd" in the directory "etc" of your Alephino installation
- a script "savepool.sh" in the directory "bin" of your Alephino installation

If you use others than the standard user/password and/or more pools you have to edit it in file savepool.cmd

To execute this procedure e.g. every day, define a Cron-Job as follows (**Note: paths must be adjusted!**):

Example: daily at 0: 30 am

- login as owner of the Alephino installation
- \$ crontab -e
- Enter the following lines:

```
30 00 * * * cd /home/exlibris/alephino/bin/; ./savepool.sh
```

After saving the entry, the job is active.

The output will be directed into the file **backup/saveprot.txt**. If the backup failed e.g. because the database was locked, the log file **DDMARC.log** in the directory **pools** will not be emptied (since release backup will not be executed). A typical output in the protocol would be in this case:

```
CMD> JOB 000005 SAVE 2014/06/16 14:59:15 START
2014/06/16 14:59:15 POOL=M PATH=
Warning: Pool has not been saved since 07.06.2014
JOB 000005 SAVE 2014/06/16 14:59:15 CANCELLED: 0022
SAVE POOL=M RELEASE=Y : 22 - Saving not possible (pool damaged?)
```

In case of success, a typical output would be:

```
CMD> JOB 000004 SAVE 2014/06/16 14:26:10 START
2014/06/16 14:26:10 POOL=M PATH=
Backup identifier file ../backup/DDMARC.SID
Pool backup          ../backup/DDMARC.S00
2014/06/16 14:26:10 RELEASE POOL=M
JOB 000004 SAVE 2014/06/16 14:26:10 END
```

1.2 Windows

In the standard, the distributed version already includes the following files:

- a command file "**savepool.cmd**" in the directory "**etc**" of your Alephino installation. If you use others than the standard user/password and/or more pools you have to edit it in this file.

1) If your Alephino server is installed as service, please create script „**save.bat**“ in the directory „**bin**“ of your Alephino installation:

```
@echo off
date /T >> ..\data\saveprot.txt
time /T >> ..\data\saveprot.txt
net stop "Alephino 5.0 Server" >> ..\data\saveprot.txt
adm_batch < ../etc/savepool.cmd >> ..\data\saveprot.txt
net start "Alephino 5.0 Server" >> ..\data\saveprot.txt
echo on
```

2) Executing this file shuts down the Alephino server and processes the command file "**savepool.cmd**". This procedure can now be used e.g. in the Window scheduler "at" to shut

down the Alephino service at a specific time, save and release the database and then restart the Alephino server. The script needs to be performed from Alephino directory **bin**. If you want to use the script from anywhere else you have to add absolute paths.

3) The output will be directed into the file `../data/saveprot.txt`. In case of success, a typical output would be:

```
16.06.2014 14:26
```

```
Alephino 5.0 Server was successfully stopped.  
Alephino 5.0 Batch Service
```

```
CMD>
```

```
CMD> JOB 000004 SAVE 2014/06/16 14:26:10 START  
2014/06/16 14:26:10 POOL=M PATH=  
Backup identifier file ../backup/DDMARC.SID  
Pool backup          ../backup/DDMARC.S00  
2014/06/16 14:26:10 RELEASE POOL=M  
JOB 000004 SAVE 2014/06/16 14:26:10 END
```

```
CMD> Trying to start Alephino 5.0 Server.  
Alephino 5.0 Server was successfully started.
```

If the backup failed e.g. because the database was locked, the log file `DDMARC.log` in the directory **pools** will not be emptied (since release backup will not be executed). A typical output in the protocol would be in this case:

```
16.06.2014 14:59
```

```
Alephino 5.0 Server was successfully stopped.  
Alephino 5.0 Batch Service
```

```
CMD>
```

```
CMD> JOB 000005 SAVE 2014/06/16 14:59:15 START  
2014/06/16 14:59:15 POOL=M PATH=  
Warning: Pool has not been saved since 07.06.2014  
JOB 000005 SAVE 2014/06/16 14:59:15 CANCELLED: 0022  
SAVE POOL=M RELEASE=Y : 22 - Saving not possible (pool damaged?)
```

```
CMD> Trying to start Alephino 5.0 Server.  
Alephino 5.0 Server was successfully started.
```

In this case it is necessary to perform “Restore database” in order to bring the database(s) back to a proper state. After this it will pass the internal check routines to be accepted for regular backup again.