

How to migrate from embedded-db to Microsoft SQL Server 2005 in OpenFire

3.6.4

By Scott Autry

Our company started out with embedded-db in OpenFire to test the platform and soon found it sufficient for our needs. Our needs grew to being able to create reports of user activity and quickly found that parsing the embedded-db was nigh impossible apart from crazy regular expressions and lots more work that we just didn't have the time for. We needed to keep our existing chat archive history along with the other settings in place. So I began looking into a way to migrate our data without a wholesale re-install. Here is what I found, step-by-step. This tutorial assumes you have a basic working knowledge of Microsoft SQL Server 2003/2005/etc and Windows Server products.

Step 1. Shut down OpenFire. Start it back up, and then shut it down again. This will insert all of your recent activity from the clients to the `openfire\embedded-db\openfire.script` file which we will use later. The majority of your work will need to be done while OpenFire is offline mainly because messaging will still be running and logging data into the `openfire\embedded-db\openfire.script` file otherwise.

Step 2. Create a database in SQL Server and run the script to create the database tables located at: `openfire\resources\database\openfire_sqlserver.sql`

Make sure at this point you know your dba username and password to grant OpenFire access during the pseudo-fresh install we will undertake in a moment.

Step 3. Make a copy of the entire `openfire` folder and keep it as a backup.

Step 4. Edit the `openfire\conf\openfire.xml` file in this way.

Find the code that says `<setup>true</setup>` and change it to `<setup>>false</setup>`

This step essentially tells OpenFire that the next time you start up the Administrator console to begin the setup wizard for a fresh install. Don't worry all of your info is still saved in the `openfire.script` file.

Step 5. This step is probably the most tedious step of them all. Here is where you are actually going to take the `openfire\embedded-db\openfire.script` and format it slightly to make it ready for a nice, smooth import into SQL. If you are not familiar with SQL statements then this might seem daunting. The reason we need to edit the file is the fact that there are many SQL statements in the `openfire.script` file that are not compatible with MS-SQL because they were written for MySQL. Here's what I remember.

- `CREATE MEMORY TABLE` will need to be `CREATE TABLE`
- `LONGVARCHAR` will need to be `NTEXT`

These lines can be deleted:

```
CREATE SCHEMA PUBLIC AUTHORIZATION DBA

CREATE USER SA PASSWORD ""
GRANT DBA TO SA
SET WRITE_DELAY 20
SET SCHEMA PUBLIC
INSERT INTO OFUSER VALUES('admin','admin',NULL,'Administrator','admin@example.com','0','0')
```

Here is where a good text editor like [Notepad++](#) comes in.

Open the `openfire\embedded-db\openfire.script` file and use the Find & Replace feature and search for `CREATE MEMORY TABLE` and replace it with `CREATE TABLE` in the editor. Repeat the process for `LONGVARCHAR` and replace it with `NTEXT`.

Step 6. Copy (Ctrl+A then Ctrl+C) the contents of the `openfire.script` file and run it on your SQL server.

CAUTION: IF YOU HAVE BEEN RUNNING OPENFIRE FOR QUITE A WHILE THE AMOUNT OF LINES YOU WILL HAVE TO COPY MAY BE IN THE TENS-OF-THOUSANDS. DEPENDING ON YOUR SQL SERVER PERFORMANCE YOU MIGHT WANT TO BREAK THE UPDATE STATEMENTS INTO SMALLER CHUNKS (I.E. COPY UPDATE STATEMENT LINES 100-1000 AND RUN THE UPDATE, ETC.) IF YOU HAVE A DECENTLY POWERFUL SERVER THEN RUN THE ENTIRE THING...JUST KNOW YOUR SERVERS PERFORMANCE AS IT MIGHT TAKE A MINUTE TO COMPLETE.

As you run the script you might notice that there are some errors stating that there are already tables that exist, simply go to the `openfire.script` file and delete that line along with its `CREATE INDEX` reference line which should be located directly below the `CREATE TABLE` line. Repeat the process until you have all of the `CREATE TABLE` scripts and the correlating `CREATE INDEX` scripts in your file deleted that already exist in the database.

I decided to write the migration manual in this way because I didn't want to any tables to be left out and since every configuration is a little different there might be some tables that are in your installation that are not in mine. Otherwise, we could have just run the `openfire.script` file to create all of the tables instead of the prior **Step 2** where we created the tables using the `openfire_sqlserver.sql` script. Make sense?

Step 7. If you've gotten this far and you received a message stating that your query ran successfully then we are done! Start OpenFire, go to the Administrator console and login just like you did before with the same credentials.

As a side note if you see the error `Violation of PRIMARY KEY constraint 'ofRRDs_pk'. Cannot insert duplicate key in object 'dbo.OFRRDS'` don't worry about it.