

Migrating MiDatabank Version 2 to Version 3

Please note: MiDatabank version 3 requires Microsoft SQL Server 2005 or Microsoft SQL Server 2008. Version 3 is more complex and resource-hungry than version 2 and may require upgrading server or client hardware to provide acceptable performance. This is especially true if any of the hardware is getting old or of low specification.

A dry-run of the migration is recommended to flush out any possible performance issues.

The migration of MiDatabank version 2 to version 3 involves installing new client software on local PCs. It also involves a brand new database format, and it is necessary to migrate data from the existing MiDatabank database (usually called mi.mdf) to the new database (a database called mi3.mdf).

Configuration of client software:

Before migrating the database, it is suggested that the MiDatabank 3 software is installed on client machines.

Once the client software has been installed, it is necessary to change the connection string so that it uses the mi3 database that will be created in the next step. The file to change is called init.xml and is contained in the following directory:

C:\Program Files\CoAcS\MiDatabank 3

The following is an example connection string:

```
packet size=4096;integrated security=SSPI;data source=pluto;persist security info=False;initial catalog=mi3
```

This connection string assumes that you have a database server called **pluto** hosts the **mi3** database

Migrating the Database

The database migration involves running a sql script that transfers data from the old database (mi.mdf) to the new database (mi3.mdf)

This process is as follows:

1. A detached database called mi3 is provided by CoAcS. The database administrator attaches this database on the SQL Server. A virgin database called mi3 is created.
2. Users are told to stop using the version 2 software.
3. A backup is taken of the current live database - this is the database called mi.mdf used by version 2. A copy of this backup should be placed in long-term storage.

4. A conversion script is provided by CoAcS. This conversion script is run on the database server. It transfers all the necessary data from mi.mdf to mi3.mdf. The script assumes that the mi and mi3 databases are on the same SQL Server Instance. If the mi3.mdf database is being hosted on a new server, then the backup of mi.mdf from 2 above should be restored on the new server prior to running the conversion script.

It is recommended that a 'dry-run' be done of the migration process above. This would involve omitting step 2 above. During the dry-run, version 2 continues to be used for live enquiries. However, the dry-run allows users to create 'pretend' enquiries using version 3, and exploration of the new features and facilities.

An important aspect of the dry-run is to identify any issues before going live. Version 3 is more complex and resource-hungry than version 2 and there may be speed issues, especially if the server and client machines are getting old.

Any data entered in the dry-run is effectively dead data and will be lost. At a later date, the dry-run database should be deleted and the process above repeated for the actual go live.

Configuration of database:

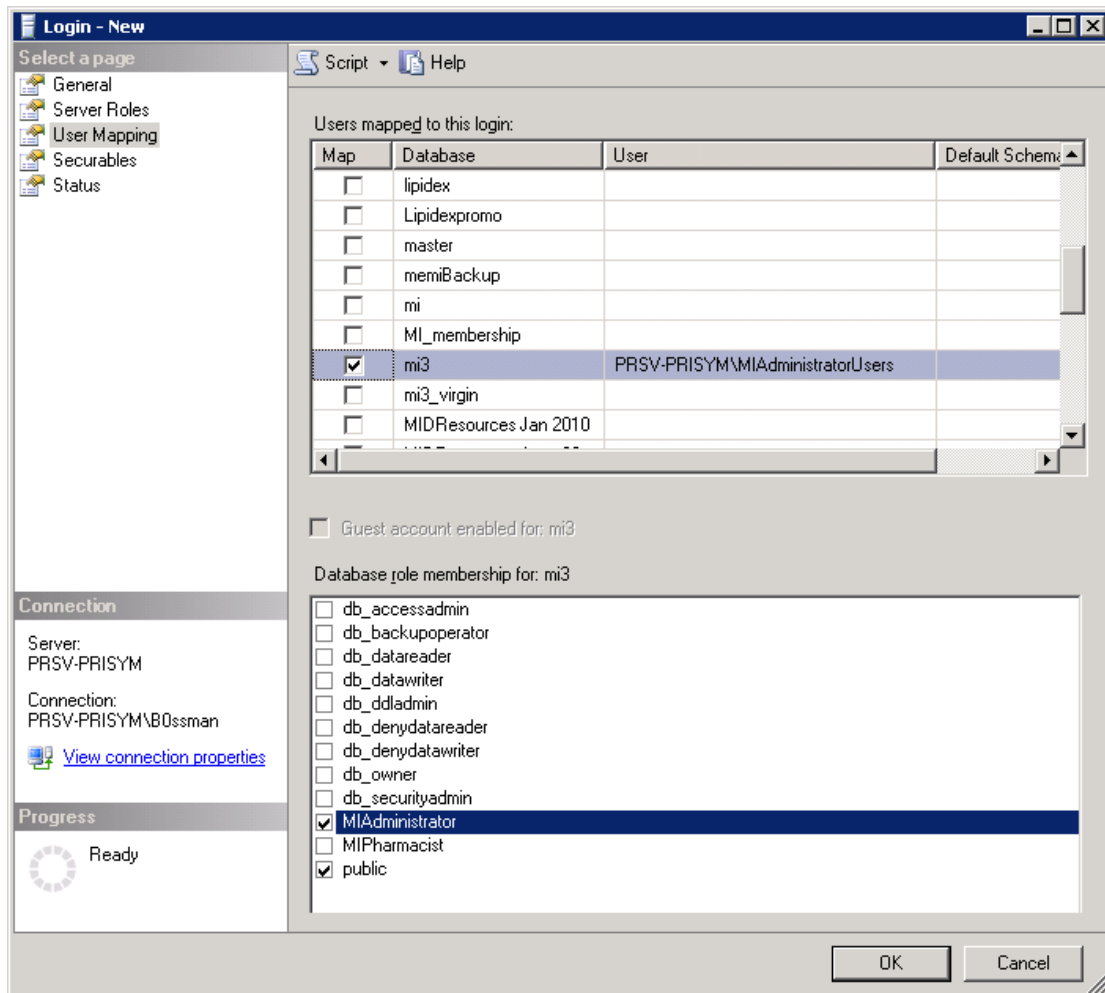
It is expected that there will already be two windows groups on Active Directory (or the local SQL server) called:

- MIAdministratorUsers
- MIEnquiryManagerUsers

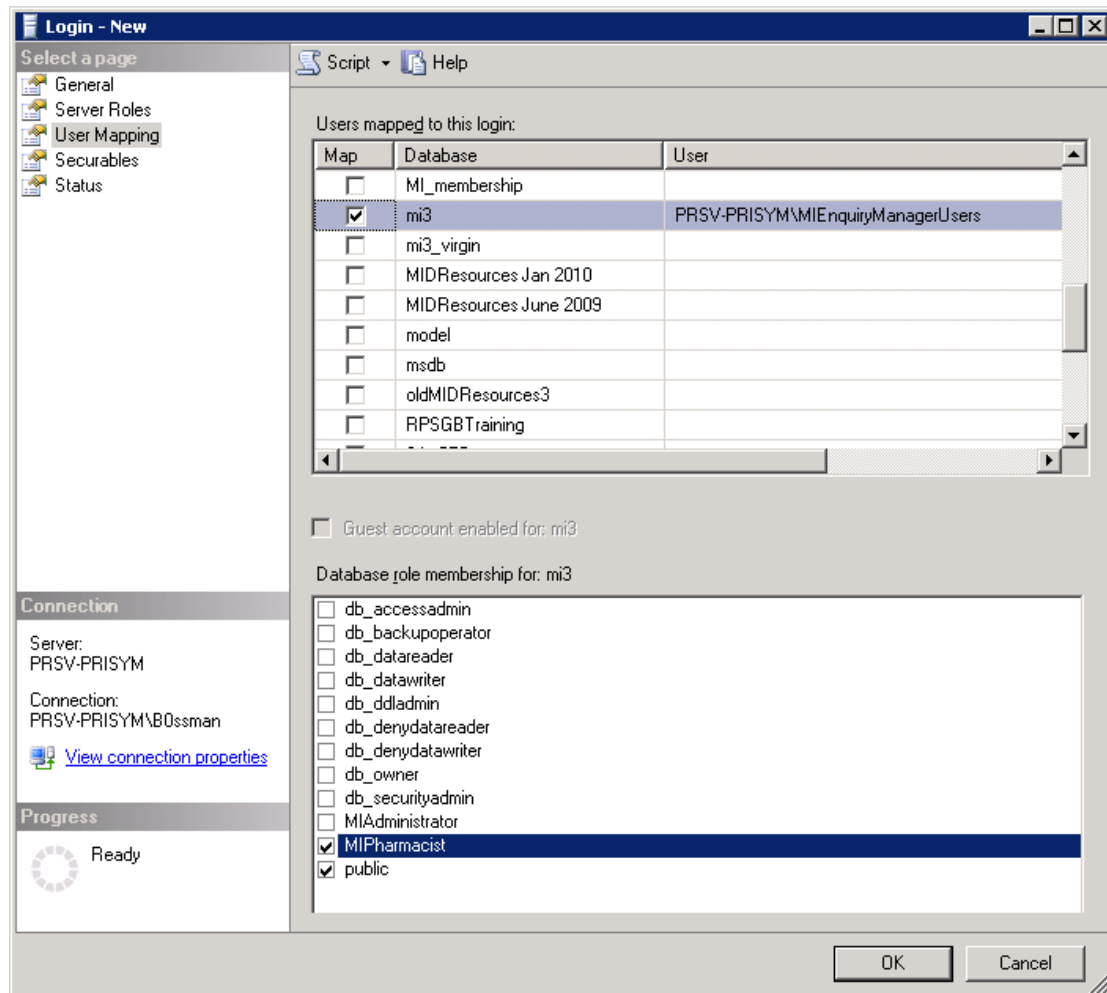
These will have been created as part of the original installation of the mi.mdf database from version 2. If these do not already exist, they should be created and populated with the appropriate windows users.

In order to complete configuration of the mi3 database, it is necessary to add the above groups as logins to mi3.mdf, and then:

For the MIAdministratorUsers login, add the MIAdministrator database role:



For the MIEnquiryManagerUsers login, add the MIPharmacist database role:

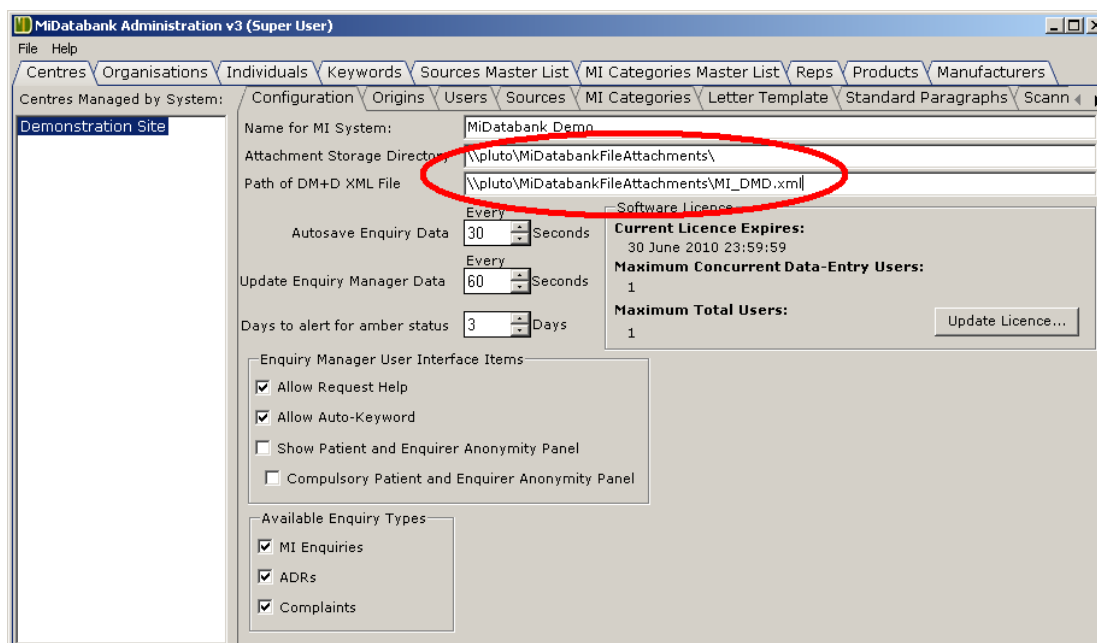


Launch the MIAdmin application: the last job is to launch the MIAdmin application and finish configuration.

Once you have launched MIAdmin, please ensure that the path for the attachment storage directory is valid. This path will be the same as used for version 2.

Migrating MiDatabank v2 to v3

The following shows an example:



If you need to change the path of this directory (e.g. because the existing server is being retired), you will need to move the contents of the existing directory to the new directory.

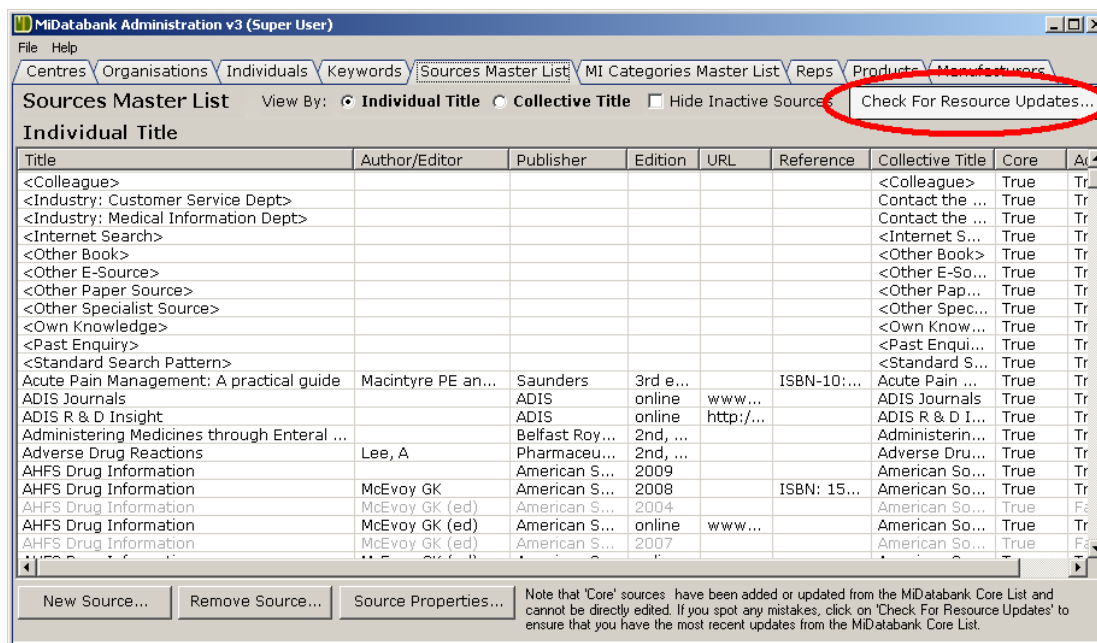
This will ensure that any attachments made using version 2 will still be available.

Enable updates to the core resource list

The main advantage of using the update mechanism is that it saves end-users a huge amount of time - there are hundreds of resources on the centrally maintained core listing.

Using the update facility means that users do not have to manually add all the commonly used resources, but can simply check for updates every month or so.

The update mechanism is contained with the MiDatabank Administration application as shown below:



Clicking on the 'Check For Resource Updates...' causes a request to be made to a web service on the internet, and data to be returned to the MiDatabank Administration application.

Please note that:-

- The data returned is limited to textual information about books and urls. If the user chooses to accept the selected information, the local database (the mi3 database as configured above) will be updated with this information.
- The data passed is one-way - from the update web-service to MiDatabank.
- The web-service has been endorsed by UKMI
- The web-service is provided by CoAcS - a UK company which is currently working towards ISO 27001 and N3 connectivity.

Since most organisations have some kind of internet security in-place to prevent incoming viruses or outgoing leakage of sensitive data, it may be necessary to make changes to the organisation's firewall rules, or otherwise allow the web-service request to succeed.

For IT departments there is a guide called '*Web Service Configuration*' available in the Documents>Installation section of the midatabank.com web-site.