Guide to migrating AIMS from one MS SQL Server to another

Version 4.3

March 2020

The following instructions are for moving your AIMS SQL Server database from one MS SQL Server version to another.

Contents

[Before You Begin 2](#_Toc348650212)

[Preparing the new SQL Server 2](#_Toc348650213)

[Migrating your data 8](#_Toc348650214)

[SQL Server Logins and Users 12](#_Toc348650215)

[Configuring the AIMS Program 16](#_Toc348650216)

[Managing AIMS users with Windows Authentication 16](#_Toc348650217)

[Troubleshooting AIMS Migration 18](#_Toc348650218)

# Before You Begin

There are four main steps to the migration process:

1. Backup the AIMS database on your current Microsoft SQL Server
2. Restore the AIMS database on the new Microsoft SQL Server
3. Grant your users permission to access the AIMS database on the new Microsoft SQL Server
4. Change the AIMS configuration file to point to the new SQL Server

While you carry out the migration, you should stop users from entering data in the old database.

To make sure that no users are using the old database, you can detach the database. This does not delete any data, just makes the database inaccessible. That way you can be sure none of your users are still entering data in the old database.

# Preparing the new SQL Server

This section summarises some of the SQL Server configuration options required for a successful AIMS installation. The screenshots are taken from two tools – ‘SQL Server Configuration Manager’ and ‘SQL Server Management Studio’. If you are using an Express version of SQL Server you may not have SQL Server Management Studio installed, but you can download it free from Microsoft.

If you are migrating from SQL Server 2000, ‘SQL Server Configuration Manager’ replaces three programs – ‘Service Manager’, ‘Server Network Utility’, and ‘Client Network Utility’. Likewise, ‘SQL Server Management Studio’ replaces the old ‘Enterprise Manager’ tool. The new tools work in a very similar way to the old tools, so you should be able to follow the instructions without difficulty.

**SQL Server Instance Name**

Launch the SQL Server Configuration Manager, select ‘SQL Server Services’ in the left hand panel, and check to see if the SQL Server state is ‘Running’. Start Mode should be set to ‘Automatic’. The name in brackets, in this case ‘SQLEXPRESS’, is the instance name. You will need this name, and the name of the machine on which the SQL Server is running, in order to connect to the database.



You do not have to have the SQL Server Browser service running if you are using the default TCP/IP port settings for SQL Server. However, if your clients are unable to connect to the SQL Server, start the SQL Server Browser service and try again.

**Network Protocol**

AIMS clients can use the TCP/IP protocol to communicate with the SQL Server, and in most network environments this is the fastest option. Ensure that this protocol is enabled for both the server and the clients. By default, in more recent versions of SQL Server, this protocol is turned off. Enable the protocol by right-clicking on the protocol name and selecting ‘enable’.



You also need to enable the TCP/IP protocol for the SQL Native Clients, again, right-click on the protocol name and select ‘enable’.



You need to stop and re-start the SQL Server for these changes to take effect. Select the SQL Server Services icon in the left-hand panel to restart the server.

|  |  |
| --- | --- |
| j0433793 | You do not need to leave the other protocols enabled. However, in earlier releases of SQL Server 2005 there was a bug that meant if you only enabled TCP/IP, although it appeared enabled it was not. The work around was to enable all protocols, re-start the server, then disable the unwanted protocols and re-start again. |

**Firewalls**

Because AIMS uses TCP/IP to communicate with the SQL Server, you need to ensure that the firewall software on your client and server machines is allowing communication to take place on the SQL Server port. Refer to the SQL Server documentation for more information.

**64-bit Windows Versions**

If you are using a 64-bit version of Windows, ensure that you install and configure the 32-bit versions of the SQL Server native client. Because AIMS is a 32-bit program it uses the 32-bit client to communicate with the server.



**Windows Authentication**

Lasa strongly recommends that you use windows authentication when using SQL Server for greater security. You can use the user administration options in the AIMS software to link AIMS users to Windows logins.

If you choose not to use Windows Authentication, then your SQL Server must be configured to use ‘SQL Authentication’. In Microsoft SQL Server Management Studio, right-click your SQL Server instance and select ‘Properties’.



In the left-hand pane of the properties window, select the ‘Security’ page. In the top section, Server Authentication, select the option for SQL Server and Windows Authentication mode. In this mode, you will be able to create Logins which are not linked to a Windows user, and from these logins you can create AIMS users.



**Language Settings**

By default, SQL Server uses American English as its language and AIMS will work with this setting. If you change the language setting, ensure that you use the same setting for both the SQL Server and the client connections. If you do not, you may find that date values have the day and month transposed when they are saved to the database. To check the setting, in Microsoft SQL Server Management Studio, right-click your SQL Server instance and select ‘Properties’.



In the Server Properties window, on the General page, you will see what language the SQL Server is using.



Now check the language that your Logins are using. In SQL Server Management Studio, go to the Logins section, under the Security heading and select a login. Right-click on the login and select Properties.



At the bottom of the login properties page you will see the setting ‘Default Language’. This should be the same as the SQL Server language, above. In this instance, “English” means American English. There is another setting for “British English”.



# Migrating your data

To backup and restore data in SQL Server, you will need administrator permissions.

On your old SQL Server, use the SQL Server Management Tool to make a backup of your AIMS database. Find the AIMS database – if you are not sure of the name, check the AIMS.ini file for your existing, working AIMS installation.

Right-click on the AIMS database, and select Tasks and then Back Up…



Make sure you select a full backup, and make a note of name of the backup file on disk, as shown in the destination part of the screen. Refer to the Microsoft documentation for details of all the other options.



Now switch to the new SQL Server, where you will restore the backup.

In the SQL Server Management Tool, right-click on Databases and select Restore Database…



In the Source section, select “Device” and click the “…” button to select the backup file.



When you click the “…” button next to “Device”, this dialog will appear. Click Add and select the backup file you created earlier. Then click OK.



Check the Files page to make sure that your new AIMS database will have the filename you expect. If you are changing the name of the database, you will need to update the filename.



Finally, check the Options page. If you are restoring the backup over an existing AIMS database – perhaps because you had a trial run – you need to tick the Overwrite option.

BE CAREFUL – ONLY TICK THIS OPTION IF YOU ARE CERTAIN YOU WANT TO OVERWRITE AND LOSE THE EXISTING DATABASE.

Click OK and when the restore is complete, you should see this message.



# SQL Server Logins and Users

It is important to understand the difference between Logins and Users in SQL Server. In the screen below, note that we have selected the ‘Security’ folder at the first level in the hierarchy. This is where Logins are located.

Logins are Windows accounts, and these should already exist on your network. These are the accounts your users use to log in to Windows. The login may have two parts – a domain and an account, in the example below “WIN8DESKTOP\Robert” represents the user account “Robert” on the domain “WIN8DESKTOP”.

Network administrators can create a group, for example “AimsUsers”, to represent all Windows accounts that can access AIMS. This makes life much simpler because you only need to grant database permissions to that group, rather than each individual account.



If your AIMS user’s account, or the AIMS user group is not listed, right-click on Logins, and select “New Login…”. Then enter or select the account or group.

When the Login has been added, right-click on it and select Properties. Then click on the User Mapping page. This is where you give the Login permission to use the AIMS database. This will create a new User in the AIMS database.



Note that in the top section “Users mapped to this login” the AIMS database is ticked. In the bottom section, the “db\_owner” database role is selected.

AIMS has functionality to modify table structure and to add or remove users built in, but for this functionality to work, you must specify ‘db\_owner’. Network administrators who object to granting this level of permission can specify ‘db\_datareader’ and ‘db\_datawriter’ instead, but some administration options in AIMS will no longer work.

**Checking the Database Users**

The above steps should create a new user for the AIMS database. You can check this by looking in the Security folder belonging to the AIMS database.

In the screen below, you can see that we have opened the Databases folder, selected the AIMS database, and under that folder we have located the security folder. Note that the folder underneath it is called ‘Users’, not ‘Logins’.



Right-click on a user to display the properties dialog. The screen below shows the properties dialog for that AIMS user. Note that the default schema should be ‘dbo’ and in the Database Role membership section, select ‘db\_owner’.



**Without Windows Authentication**

In a non-Windows Authenticated system, you need to create a new SQL Server Login for each AIMS user. The Login must have the prefix ‘AIMS\_’. The suffix is the User ID used inside the AIMS software itself. For example, AIMS is installed with a default User ID ‘ADMIN’. The Login and User in SQL Server that is used by this AIMS user is ‘AIMS\_ADMIN’. The prefix is used to distinguish AIMS users from any other logins in the SQL Server. Each Login is given the same password, ‘AIMS’. Clearly this is insecure, and this is why we recommend using Windows Authentication instead.



Connect these logins to the AIMS database as described in the previous section.

#

# Configuring the AIMS Program

If you are migrating from one SQL Server to another, you should already have a working network installation of AIMS. You just need to modify the AIMS.ini file in order to tell AIMS to use the new SQL Server.

The AIMS.ini file is located in the same directory as the AIMS program (AIMS.exe). Open the AIMS.ini, using Notepad or another plain text editor.

This file tells the AIMS.exe program where the database is located, and what it is called. The entry should look something like this:



The first line should say “Server=<Machine>\<Instance>” where <Machine> must be the correct name for the machine where SQL Server is running, and <Instance> is the name of the SQL Server instance where the AIMS database is installed. In the above example, the machine is ‘ROBERTDESKTOP’ and the instance is ‘SQLExpress’. For more information about instance names, see the section on SQL Server Configuration, above.

The line “Database=” must have the correct name of the AIMS database in SQL Server. By default this is ‘AIMS’ but it is possible that your installation may be different.

If you are using Windows Authentication, as recommended, then the final line should say “WindowsAuthentication=True”. If you are not using Windows Authentication, then change this line to “WindowsAuthentication=False”. For more information about windows authentication, see the section on SQL Server Configuration, above.

Save any changes you make to the AIMS.ini file, taking care that Notepad does not save the file as “AIMS.ini.txt”. There is no need to re-start the SQL Server, just re-start AIMS on a client machine.

# Managing AIMS users with Windows Authentication

If you were not using Windows Authentication on your old SQL Server, but you are on the new one, you can link your AIMS user details to Windows login details, so that users do not need to enter a password when they start using AIMS. To have the user name supplied automatically, you need to tell AIMS what each user’s Windows login is.

Using the Manage Users option in the AIMS configuration menu, edit each user. You should see a new edit box for the Windows Login – see below. Note that the Windows Login is case sensitive.

If you cannot see the Windows Login field, then you may not have Windows Authentication enabled. Check the AIMS.ini file – see the section Configuring the AIMS Program, above.



Note that, even if the user is entering the correct user name and password in the AIMS login screen, if their Windows login does not have permission to use the AIMS database on SQL Server, they will not be able to connect to the database.

You can create new users from within the AIMS configuration menu, but you will need to supply the login name and password for a SQL Server login with ‘sysadmin’ rights on the SQL Server. See the section on Preparing SQL Server, above, for more details.



Even though users do not need their AIMS passwords to log in to AIMS when using Windows authentication, if the user logs in and then uses the Login option from within AIMS, they can log in as another user. For better security, Users should still change the AIMS passwords from the default ‘password’.

# Troubleshooting AIMS Migration

|  |  |
| --- | --- |
| j0433793 | If you have any problems setting up AIMS, please read this section before contacting Lasa. |

The AIMS network installation is designed to be as simple and flexible to install and maintain as possible. The AIMS software, manuals and client installation programs are installed on the network file server so they are accessible from each client workstation.

Client machines get a link to the AIMS program file (AIMS.exe) on the server – this means that if you need to upgrade the program at some future time, you only need to do it in one place. Information about the database server required to make a database connection is kept in a text file, AIMS.ini, installed in the same location as AIMS.exe.

If you have problems getting AIMS to run and connect to the database, please follow the steps below.

**On the client machine:**

1. Check that the shortcut to the AIMS.exe is pointing to the AIMS.exe file installed on your server. There should not be an AIMS.exe file on the client machine.
2. Please ensure you have run the client installation program to install the SQL Server native client appropriate to your SQL Server version.
3. If AIMS runs but is unable to connect to the database, check the AIMS.ini file on the server
4. If you are trying to connect to SQL Server without using Windows Authentication and you see the warning below, then your SQL Server is not configured to allow connections using SQL Server authentication. See the section on Windows Authentication in Configuring SQL Server, above.



**On the server machine:**

1. Check the SQL Server is installed and running
2. Check the AIMS.INI file.

Browse to the folder where you installed the AIMS program (AIMS.exe) and you should find the AIMS.ini file there. Open it using Notepad or another plain text editor. It should look similar to the example below:



The first line should say “Server=<Machine>\<Instance>” where <Machine> must be the correct name for the machine where SQL Server is running, and <Instance> is the name of the SQL Server instance where the AIMS database is installed. In the above example, the machine is ‘ROBERTDESKTOP’ and the instance is ‘SQLExpress’. For more information about instance names, see the section on SQL Server Configuration, above.

The line “Database=” must have the correct name of the AIMS database in SQL Server. By default this is ‘AIMS’ but it is possible that your installation may be different.

If you are using Windows Authentication, as recommended, then the final line should say “WindowsAuthentication=True”. If you are not using Windows Authentication, then change this line to “WindowsAuthentication=False”. For more information about windows authentication, see the section on SQL Server Configuration, above.

Save any changes you make to the AIMS.ini file, taking care that Notepad does not save the file as “AIMS.ini.txt”. There is no need to re-start the SQL Server, just re-start AIMS on a client machine.

1. Refer to the SQL Server Configuration section, above, and ensure that you have enabled the TCP/IP protocol in BOTH the network and native client areas of the SQL Server Configuration Manager. Make sure that the port you are using is not firewalled on the client or server machines.
2. If you are not using the default TCP/IP port, 1433, for SQL Server you should ensure that the SQL Server Browser service is started on the server.
3. 64-bit Windows Clients: If you are installing AIMS on a 64-bit version of Windows, you still need to install a 32-bit SQL Server client because AIMS is a 32-bit application.
4. If you notice that date values are not being saved correctly, and that the date and month are being transposed, for example 1st February 2009 (01/02/2009) is saved as 2nd January 2009 (02/01/2009), then check that the language settings in your SQL Server and the login your are using are the same – see the section on SQL Server Configuration, above. You should also check the international settings on your Windows PC and ensure the PC is using the dd/mm/yyyy date format.

End.