NotifyMDM
Mobile Device Management

NotifyMDM Database Table Maintenance

This guide provides information on . . .

. . . The NotifyMDM Database Tables

. . . Database Back Up

. . . Database Cleanup Tasks

. . . Database Migration

Table of Contents

Database Maintenance 3

Transaction Log Configuration 3
Database Cleanup 3
System Backup 3

The Database Task Scheduler 4

The NotifyMDM Database Cleanup Tasks 4
Adding a Database Task 6
Managing the Database Tasks 7
Task Parameters: Best Practices/Recommendations 8

Moving the NotifyMDM Database 11

Back Up Before You Begin 11
Setting Up the New Database Server 11
Moving the Database 12
Completing and Verifying the Move 15
Additional Steps 16

Database Maintenance

Database cleanup and backup are two key elements in maintaining and insuring efficient system performance. The best practices outlined below should be incorporated into your organization's system maintenance routine.

Transaction Log Configuration

The *NotifyMDM* database is configured to prevent the transaction log from growing too large. Simple recovery mode is enabled and the maximum size for the log is set to 16 GB. When the log approaches the maximum size, alerts are issued. High volume systems may want to monitor the transaction log to determine whether the maximum size should be set to a value higher. Notify Technology Corporation Technical Support can assist you with ways to monitor the log.

Database Cleanup

Verify that the database cleanup tasks have been enabled. When the *NotifyMDM* server software is installed, tasks are enabled, by default, with parameters for a system accommodating 1000 devices. Administrators of larger systems should adjust the task parameters according to the recommendations in the Database Task Scheduler section of this guide. To verify that the jobs are running, access the *Database Task Scheduler* from the Dashboard and view the task grid. The grid displays the cleanup jobs that are enabled, the last time each job was executed, and when each job will run again.

If a database task fails to run, you can check the *DatabaseTaskSchedulerLogs* database table for errors. See the System Administration Guide: Server Logging.

System Backup

Periodically backing up the database is an essential practice for system maintenance. A daily backup of the database, preferably streamed off site, is recommended at minimum.

In addition, back up the MDM.ini file on the Web/Http server. This file is found under the *NotifyMDM* directory. Default directory: C:\Program Files\NotifyMDM Server

Regular backups ensure that data can be recovered if the database becomes compromised. With both a database backup and a backup of the MDM.ini file, a system can be fully restored if necessary.

The Database Task Scheduler

When devices connect to the *NotifyMDM* Server, information regarding those connections is logged in the database and stored for potential troubleshooting purposes.

The amount of information that is logged depends on several factors such as, the number of users on the *NotifyMDM* Server, the type of traffic being sent back and forth, the amount of logging taking place, and the frequency of device connection intervals. Over time, this information can build up in the database and will become difficult to manage.

Administrators can use the *Database Task Scheduler* to set cleanup jobs to run at regular intervals in order to clear excess data and maintain optimal database performance.

The NotifyMDM Database Cleanup Tasks

Upon installation, *NotifyMDM* database jobs are automatically enabled with the recommended parameters for a system accommodating up to 1000 devices. Administrators of larger systems should adjust the task parameters according to the recommendations in the best practices guidelines in this document.

Tasks can be edited, added, or removed through the *Database Task Scheduler*. You can also use the task scheduler to manually initiate a task so that it bypasses the frequency interval and runs immediately.

Additional tasks can be created to run customized SQL scripts at regular intervals.

Most of the default database tasks are cleanup jobs. There is also a default job that defragments the database indexes.

Database Task Descriptions

AirProxyLogs	Cleans up device sync logs.
Alerts	Cleans up messages from the *Alerts* grid.
APNTracking	Cleans up iOS APN request logs.
DatabaseTaskSchedulerLogs*	Cleans up the database task scheduler traffic; all tasks that executed successfully or that gave an error.
DataUsageLogs	Cleans up device traffic information.
Defragment Indexes*	Reorganizes data stored in SQL database indexes in order to maintain optimal database query performance.

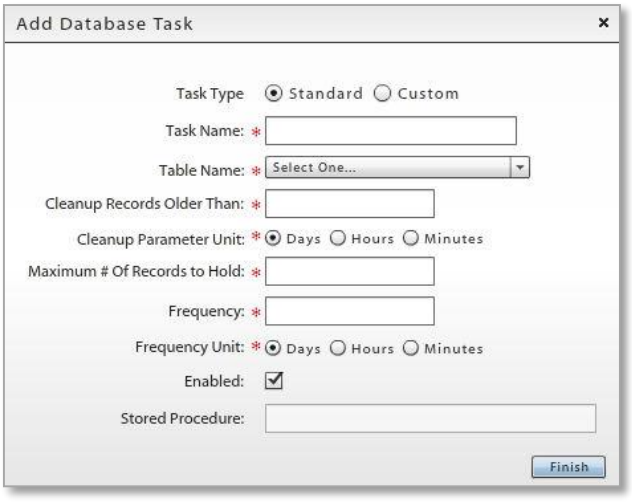
DeviceLocations	Cleans up device location data.
DeviceLogs	Cleans up logs retrieved from the device when device logging is enabled.
DeviceStatistics	Cleans up general device information.
ErrorChainLogs	Cleans up detailed error messages when an APN command does not execute successfully.
iOSDeviceInformation	Cleans up specific device information related to an iOS device.
iOSDeviceNetworkInformation	Cleans up iOS device's hardware address, phone number, SIM card and cellular information.
iOSInstalledApplications	Cleans up information about apps installed on an iOS device.
iOSInstalledCertificates	Cleans up information about all the certificates installed on an iOS device.
iOSInstalledConfigurationProfiles	Cleans up information about various configuration profiles installed on the iOS devices. When information is deleted from this table, it is also deleted from two other tables that it references: **iOSProfileSignerCertificates** and **iOSInstalledConfigurationProfilePayloads**.
iOSMDMSyncLogs	Cleans up iOS APN traffic logs.
iOSSecurityInfo	Cleans up iOS device security information, such as encryption capabilities and whether the device is passcode protected.
LicensingLogs*	Cleans up server licensing logs.
MailMessageLogs*	Cleans up records of group email sent from the Dashboard.
MDMSyncLogs	Cleans up NotifyMDM device app logs.
PhoneLog	Cleans up device phone logs.
Recover License Seats	Cleans up devices that have been in a pending delete state for 30 days, after an administrator has issued the *Stop Managing Device* command.
Rollover Logs	Removes all records in DataUsageLogs, Alerts, and DeviceAppsUsage tables if the SAkey has reached more than 2 billion.
TextMessageLog	Cleans up device text message logs.
Warnings*	Cleans up server and device warning logs.

*Data is system based; all other tasks clean up user data

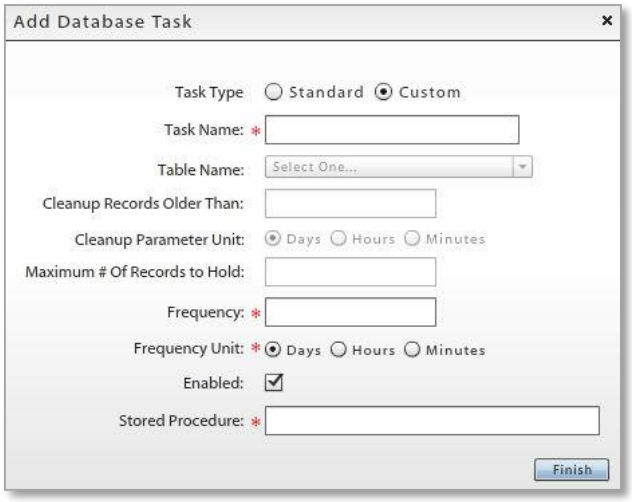
Adding a Database Task

Administrators can access the *Database Task Scheduler* through the *System Management* view of the *NotifyMDM* Dashboard. You must have full administrative login credentials.

1. Select ***System Management*** > ***System Administration*** > ***Database Task Scheduler***.
2. Click the ***Add Database Task*** button.
3. Select the ***Task Type***.
    - Choose ***Standard*** to schedule a cleanup job for one of the database tables listed in the *Table Name* drop-down list.
    - Choose ***Custom*** to schedule a stored procedure (or SQL script).



Adding a Standard Table Cleanup Task



Adding a Custom Task

4. Enter a ***Task Name***. If you are creating a *Custom* task, skip to *Frequency*.
5. Select the database ***Table Name*** from the drop-down list.
6. Enter the cleanup parameters, which define the age of the records to be deleted from the database table. In the ***Cleanup Records Older Than*** and ***Cleanup Parameter Unit*** fields, enter a number and select ***Days***, ***Hours***, or ***Minutes***.

   Example: If you enter ***7 Days***, each time the task runs it deletes records older than 7 days.
7. Enter the ***Maximum # of Records to Hold***. This puts a limit on the number of records to store in the database table. If the task is configured to keep 7 Days of information, but the records still exceed the maximum, the oldest records are deleted until the maximum is reached.
   Enter 0-999999999.
8. Enter the frequency parameters, which define how often the task runs. In the Frequency and Frequency Unit fields, enter a number and select ***Days***, ***Hours***, or ***Minutes***.

   Example: If you enter ***1 Days***, the task runs every day.
9. Check the ***Enabled*** box to activate the task.

10. If you are creating a *Custom* task, enter the name of the ***Stored Procedure***. Verify that you have entered the name correctly.

    Note: If the Database Task grid shows that a custom task has not run, it is likely that the name of the stored procedure was entered incorrectly. Click the *Run Database Task* button and select the *Last Runtime* field in the grid to test the stored procedure.

11. Click ***Finish*** to save the task.

Managing the Database Tasks

Database tasks can be added, edited, or removed using the scheduler. You can also manually initiate a task so that it bypasses the frequency interval and runs immediately.

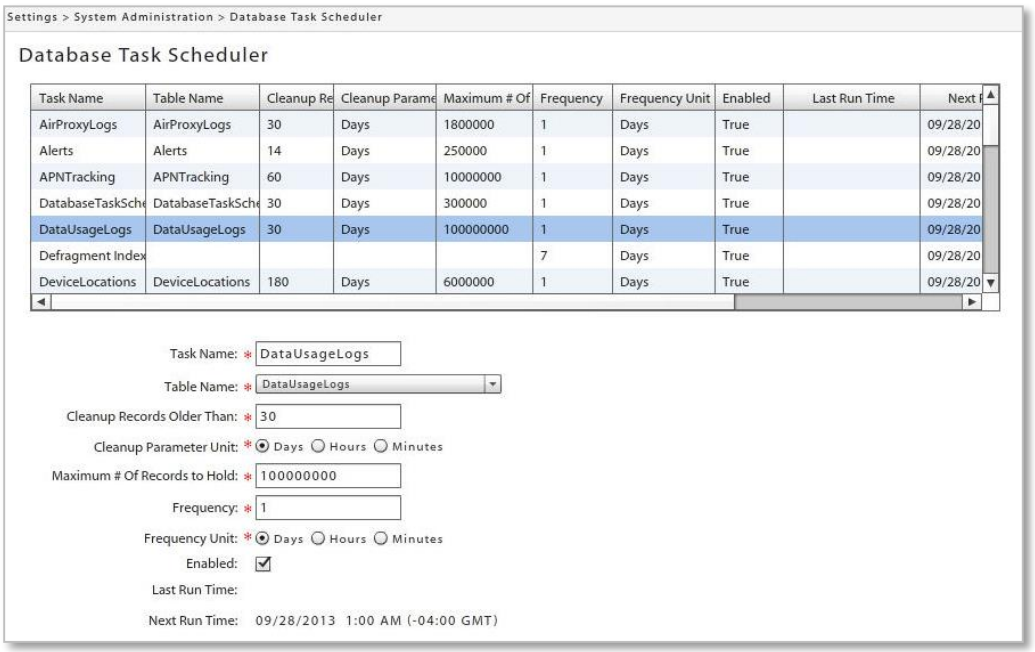
Access the *Database Task Scheduler* by selecting ***System Management*** > ***System Administration*** > ***Database Task Scheduler***.

The options bar at the top of the view includes the following management options:

- Add Database Task
- Remove Database Task
- Run Database Task
- Save Changes (select after editing a task)

The *Database Task Scheduler* displays a grid of the standard and custom tasks that have been configured.

Select a task from the grid to edit its parameters, to remove it, or to run the task immediately.



Task Name	Table Name	Cleanup Re	Cleanup Parame	Maximum # Of	Frequency	Frequency Unit	Enabled	Last Run Time	Next
AirProxyLogs	AirProxyLogs	30	Days	1800000	1	Days	True		09/28/20
Alerts	Alerts	14	Days	250000	1	Days	True		09/28/20
APNTracking	APNTracking	60	Days	10000000	1	Days	True		09/28/20
DatabaseTaskSche	DatabaseTaskSche	30	Days	300000	1	Days	True		09/28/20
DataUsageLogs	DataUsageLogs	30	Days	100000000	1	Days	True		09/28/20
Defragment Index					7	Days	True		09/28/20
DeviceLocations	DeviceLocations	180	Days	6000000	1	Days	True		09/28/20

Troubleshooting Tip: If a database task has failed to run, you can check the *DatabaseTaskSchedulerLogs* database table for errors.

Task Parameters: Best Practices/Recommendations

The following recommendations for database cleanup task parameters assume that systems have been configured according to the hardware requirements for scaling outlined in the *NotifyMDM* *System Performance: Sizing and Tuning* guide.

Database Tables that Accumulate Quickly

Because of rapid data accumulation, several tables in the database are configured, by default, to retain only the data from the last synchronization cycle of each device. The most recent values continuously replace data stored from the previous synchronization of each device. Historical data is not maintained.

Database tables that retain only the most recent values are:

- DeviceStatistics
- iOSDeviceInformation
- iOSDeviceNetworkInformation
- iOSInstalledApplications
- iOSInstalledCertificates
- iOSInstalledConfigurationProfiles *(and tables it references)*
- iOSSecurityInfo

If organizations want to maintain historical data instead of just most recent values for these tables, administrators can run a script to remove these constraints and implement the recommended job parameters (a future version of *NotifyMDM* will include Dashboard access to configuration settings). Organizations that elect to maintain historical data for these tables should reevaluate and scale their hardware configuration to accommodate the additional data.

Note: Organizations should not keep historical data for these tables when using Microsoft SQL Express.

The iOSInstalledConfigurationProfiles Table

The *iOSInstalledConfigurationProfiles* table references two other tables:

- *iOSProfileSignerCertificates*
- *iOSInstalledConfigurationProfilePayloads*

When information is deleted from *iOSInstalledConfigurationProfiles*, it is also deleted from these two tables. When only the most recent values are stored for *iOSInstalledConfigurationProfiles*, the same is true for these two tables.

The DeviceLocations Table

The *DeviceLocations* table can also accumulate data quickly when organizations elect to track device locations. If historical location data is not a necessity, this table can also be configured to store only data from the last synchronization cycle of each device. Currently, this is accomplished with a database script (a future version of *NotifyMDM* will include Dashboard access to configuration settings).

The parameters in the table below are the default settings for the database cleanup jobs, which are automatically enabled when *NotifyMDM* is initially installed.

These parameters are recommended for a system accommodating 1,000 devices. Administrators of larger systems should adjust the task parameters according to the recommendations in the best practices guidelines outlined below this table.

For tables marked with an asterisk, the default behavior is to retain only the data from the last synchronization cycle of each device; although recommended settings are given if you want to keep historical data. (Administrators can run a script to remove these constraints and implement the recommended job parameters. A future version of *NotifyMDM* will include Dashboard access to accomplish this.)

*Tasks that, by default, retain only the data from the last synchronization cycle of each device.

Task Name	Cleanup Records Older Than	Frequency	Maximum # of Records to Hold
Database Cleanup Jobs			
AirProxyLogs	30 Days	Daily	1,800,000
Alerts	14 Days	Daily	250,000
APNTracking	60 Days	Daily	10,000,000
DatabaseTaskSchedulerLogs *System data is not affected by number of devices*	30 Days	Daily	300,000
DataUsageLogs	30 Days	Daily	100,000,000
DeviceLocations	180 Days	Daily	6,000,000
DeviceLogs	365 Days	Daily	50,000
DeviceStatistics* If you keep historical data:	30 Days	Daily	1,000,000
ErrorChainLogs	30 Days	Daily	100,000
iOSDeviceInformation* If you keep historical data:	30 Days	Daily	1,000,000
iOSDeviceNetworkInformation* If you keep historical data:	30 Days	Daily	1,000,000
iOSInstalledApplications* If you keep historical data:	30 Days	Daily	60,000,000
iOSInstalledCertificates* If you keep historical data:	180 Days	Daily	12,000,000
iOSInstalledConfigurationProfiles* If you keep historical data:	180 Days	Daily	36,000,000
iOSMDMSyncLogs	30 Days	Daily	10,000,000
iOSSecurityInfo* If you keep historical data:	30 Days	Daily	1,000,000
LicensingLogs *System data is not affected by number of devices*	365 Days	Daily	500
MailMessageLogs	365 Days	Daily	500,000
MDMSyncLogs	30 Days	Daily	1,000,000

PhoneLog	365 Days	Daily	4,000,000
Recover License Seats	-	Daily	-
Rollover Logs	-	Daily	-
TextMessageLog	365 Days	Daily	20,000,000
Warnings	30 Days	Daily	250,000
Stored Procedures			
Defragment Indexes	-	Weekly	-

Systems with More than 1000 Devices

For systems accommodating more than 1,000 devices, the ***Maximum # of Records to Hold*** recommendations listed above should be multiplied to arrive at a figure that accommodates multiples of one thousand devices. You can enter values up to 999,999,999.

For example, maximum records for the *DataUsageLogs* and *AirProxyLogs* tables should be configured as follows:

DataUsageLogs Table		Maximum # of Records to Hold	Approximate Records per Device
1,000 Device System	100,000,000	100,000,000	100,000
5,000 Device System	5 x 100,000,000	500,000,000	100,000
25,000 Device System	25 x 100,000,000	999,999,999 *enforced maximum*	40,000
100,000 Device System	100 x 100,000,000	999,999,999 *enforced maximum*	10,000
125,000 Device System	125 x 100,000,000	999,999,999 *enforced maximum*	8,000

AirProxyLogs Table		Maximum # of Records to Hold	Approximate Records per Device
1,000 Device System	1,800,000	1,800,000	2,000
5,000 Device System	5 x 1,800,000	9,000,000	2,000
25,000 Device System	25 x 1,800,000	45,000,000	2,000
100,000 Device System	100 x 1,800,000	180,000,000	2,000
125,000 Device System	125 x 1,800,000	225,000,000	2,000

Click the ***Save Changes*** button in the action bar after you make edits to a task.

Moving the NotifyMDM Database

Back Up Before You Begin

- Back up the *NotifyMDM* database on the original server.
- If you will also be moving the *NotifyMDM* Web/Http Component, back up the MDM.ini file on the Web/Http server. This file is found under the *NotifyMDM* directory. Default path: C:\Program Files\NotifyMDM Server.

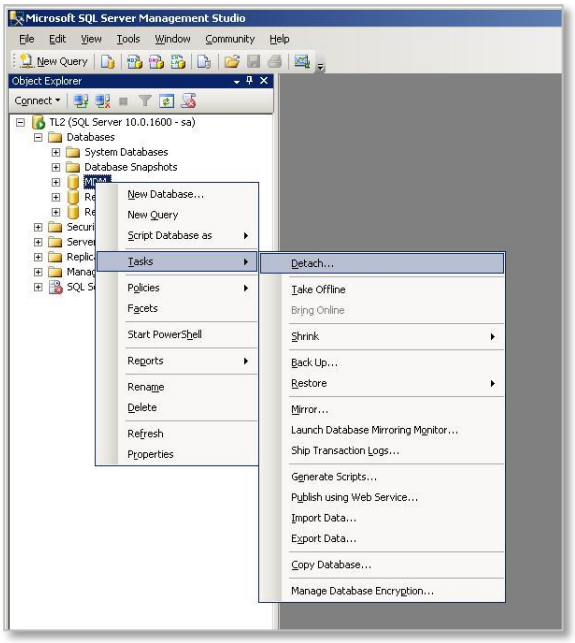
Setting Up the New Database Server

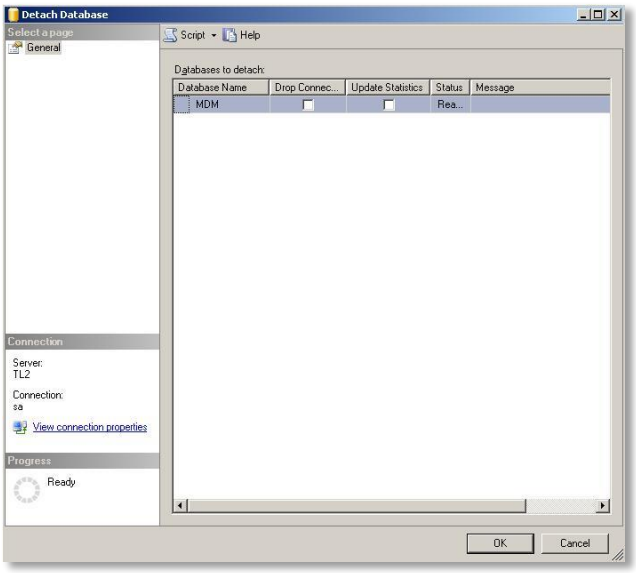
1. Fully install the *NotifyMDM* Server software to the new server. In order to perform updates on the new server, the *NotifyMDM* Web/Http component and the SQL database component must be installed.
2. Using the *NotifyMDM Update Manager*, patch the new system to the same version as the original system. Both systems must be patched to the same version to successfully complete the database move.
3. If you are moving only the SQL component and leaving the Web/Http component installed on the original machine, the Web component should be removed from the new server when the Update Manager has patched the system.

   The Web/Http component, wherever it is housed, points to the new SQL component location and facilitates any future updates.

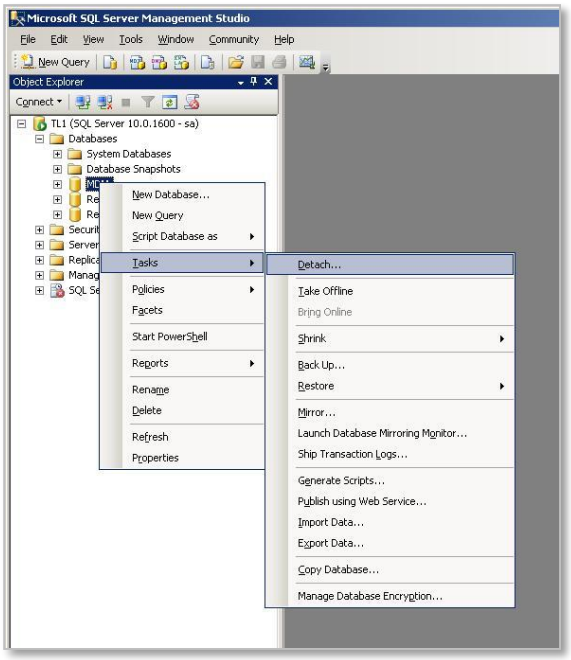
Moving the Database

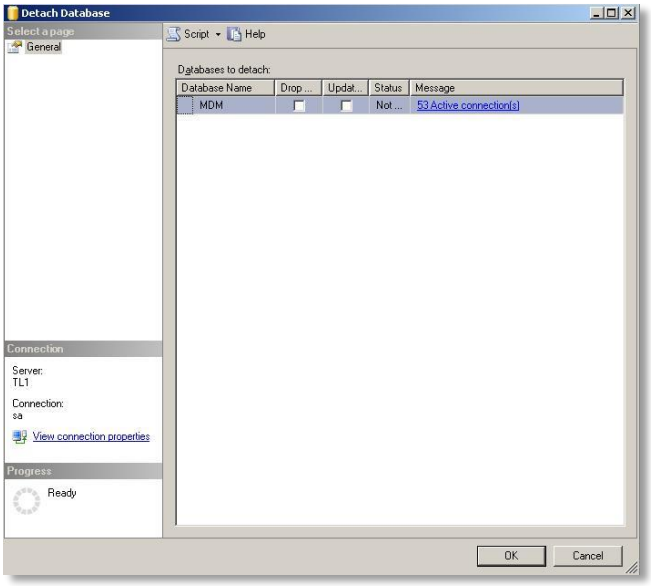
1. Stop IIS on the server where the Web/Http Component resides.
2. Detach the *NotifyMDM* database from the new server:
    a. Right-click the ***MDM*** database and select ***Tasks*** > ***Detach***.
    b. Select ***OK*** at the next screen to complete the detach.





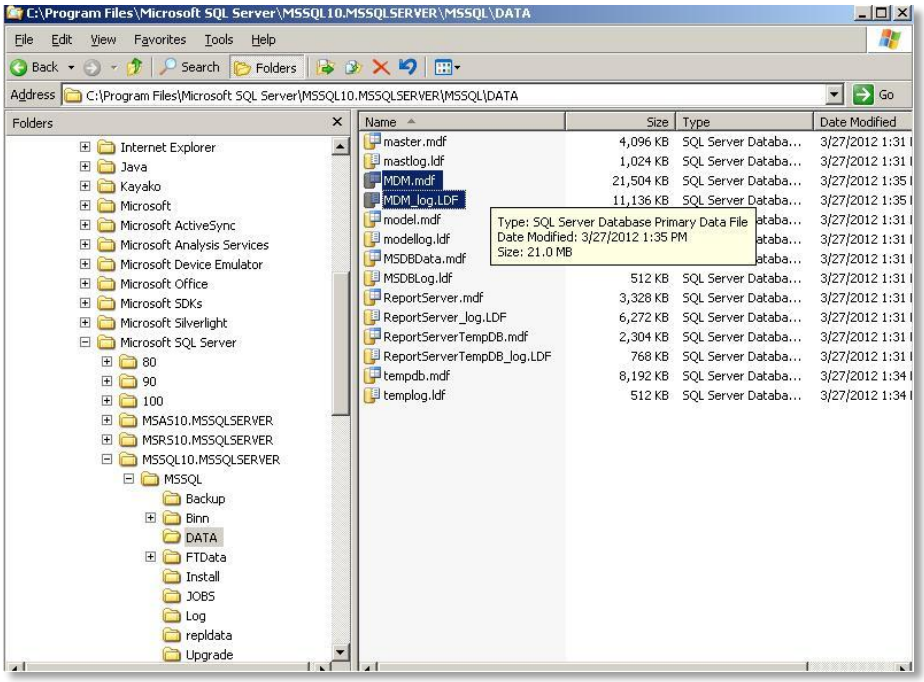
3. Detach the *NotifyMDM* database from the original server:
    a. Right-click the ***MDM*** database and select ***Tasks*** > ***Detach***.
    b. Select ***OK*** at the next screen to complete the detach.





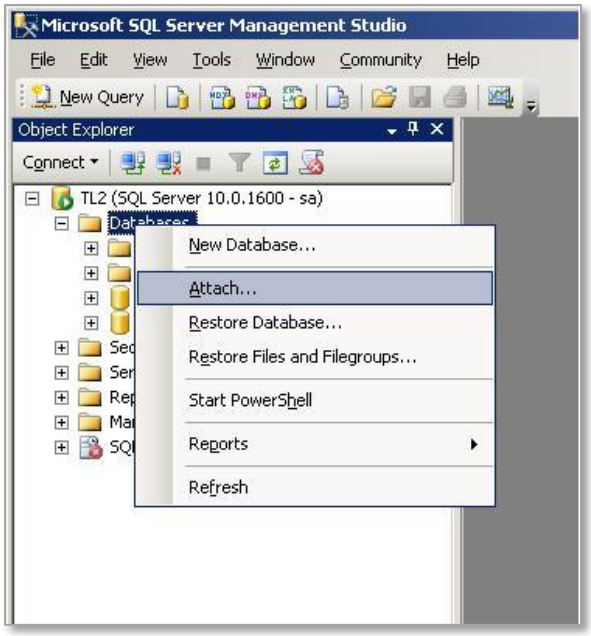
4. Move the **MDM.mdf** and **MDM_log.ldf** files from the original server to the new server:

    They are found in the directory C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA.

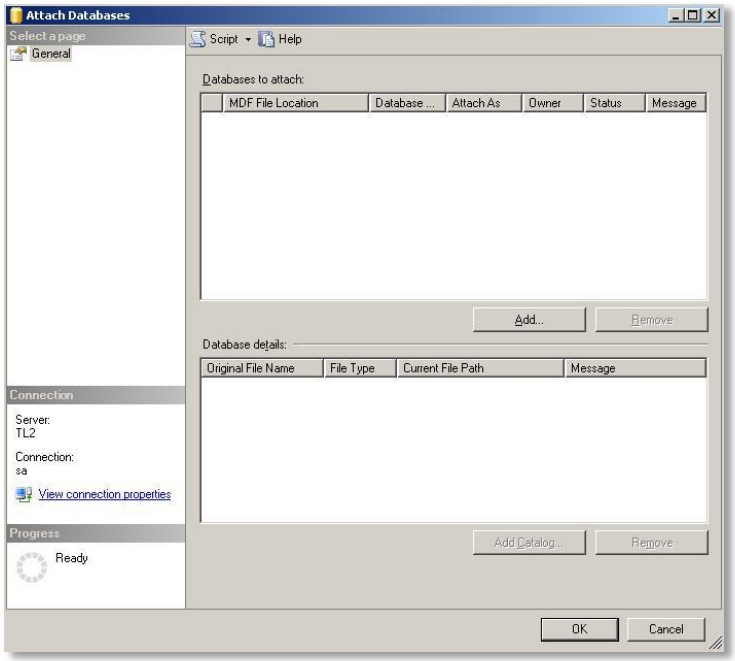


5. Attach the *NotifyMDM* database to the new server.

a. Right-click ***Database*** and select **Attach**.



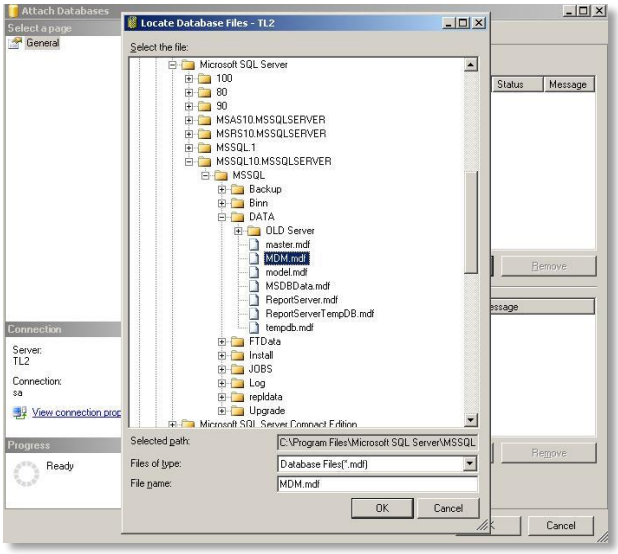
b. Select ***Add*** to add the database to the server.

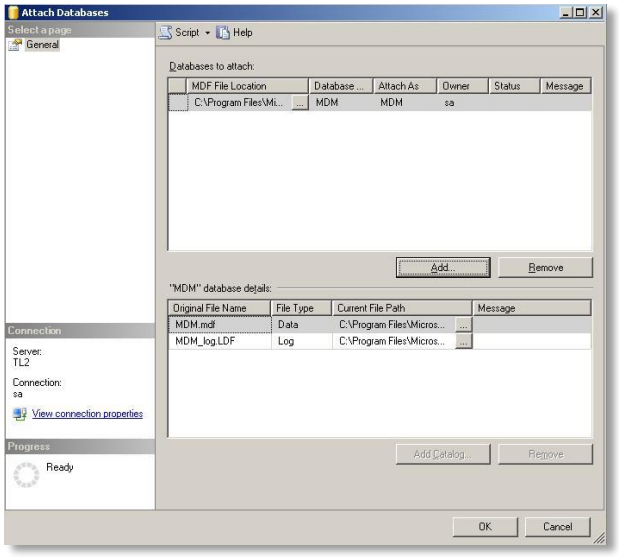


c. Locate the **MDM.mdf** file copied from the original server.

   Default directory: C:\Program Files\Microsoft SQL Server\MSSQL10.MSSQLSERVER\MSSQL\DATA

d. Select ***OK*** to complete the attach.





6. If you are also moving the Web component, copy the MDM.ini file from the original system to the new system.

7. The MDM.ini file must be updated to point to the new SQL Server regardless of whether the Web component is remaining on the original server or being moved to a new server. Contact Technical Support for assistance in updating this file.

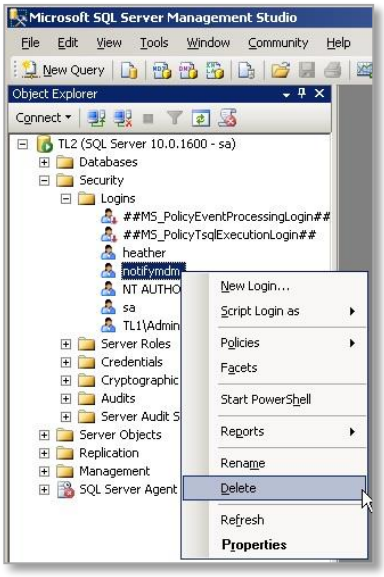
Completing and Verifying the Move

1. Start IIS on the Web server.
2. Perform an ODBC test from the Web server to ensure a successful connection to the database server.
3. Verify the new setup by logging in to the *NotifyMDM* Dashboard and ensuring that devices are actively syncing.

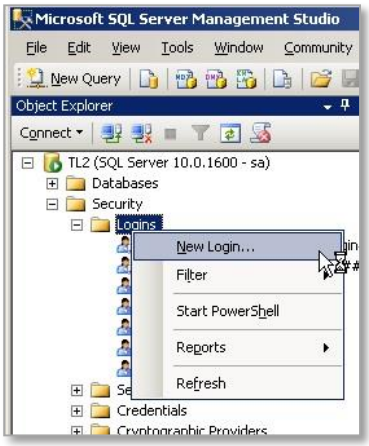
Additional Steps

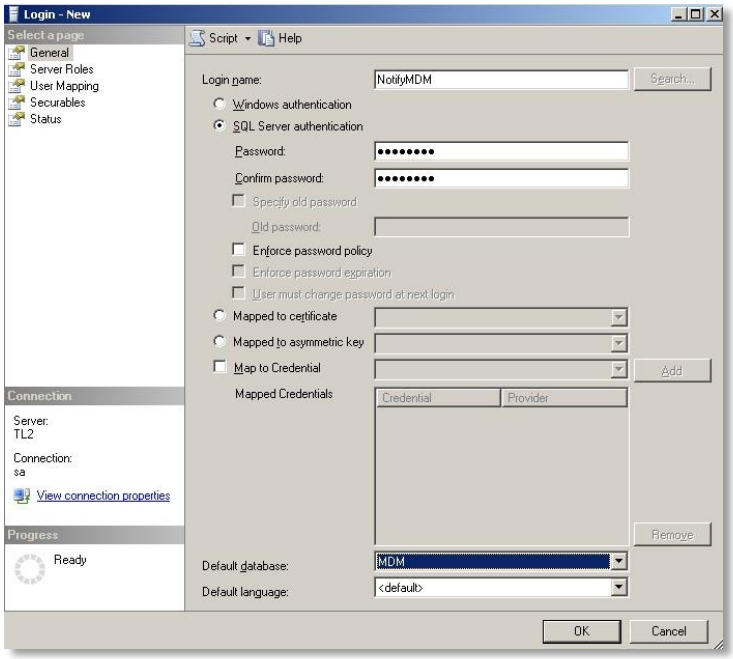
If you have completed all the steps in this section and are experiencing issues, you may need to remove and re-create the *NotifyMDM* user within SQL.

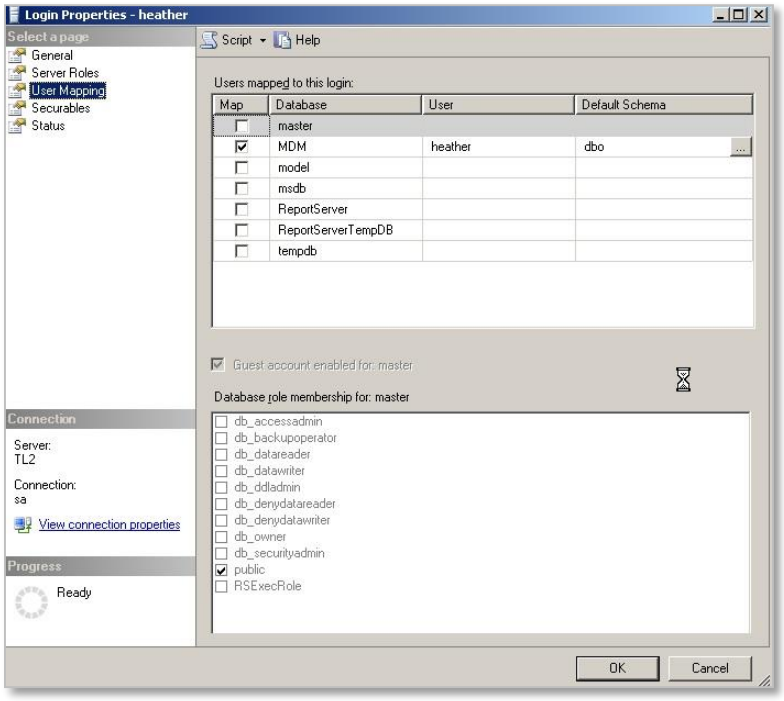
1. Delete the *NotifyMDM* user.



2. Re-create the *NotifyMDM* user.







a. Select ***SQL Server Authentication*** and set the password.

b. Deselect the ***Enforce password policy*** option so it does not change.

c. Set the ***Default Database*** to MDM.

d. Under ***User Mapping***, verify that MDM is selected.

NOTE: If you do not recall the credentials you used previously, you can create a new user. Creating a new user requires you to update the username and password within the MDM.ini file. Contact Technical Support for assistance in updating that file.