



DATABASE UTILITY USER GUIDE

June 29, 2012

2201 Thurston Circle
Bellevue, NE 68005
www.tigerpawsoftware.com

Contents

Overview of the Database Utility	3
Open the Tigerpaw Database Utility	4
Create a Tigerpaw Database	6
Attach an Existing Tigerpaw Database to your SQL Server	8
Detach a Tigerpaw Database from your SQL Server	10
Back up a Tigerpaw Database	11
Restore a Tigerpaw Backup Database	13
Shrink the Log File of your Tigerpaw Database.....	14
Clean up Inventory Data in a Tigerpaw SQL Database.....	16
Database Maintenance Performed by the Inventory Clean Up Utility	17
Clear a Lock on a Tigerpaw Database.....	22
Upload a Copy of your Database to Tigerpaw Software.....	23
Run a SQL Script on your Tigerpaw Database.....	25

Overview of the Database Utility

When you install Tigerpaw, the system also installs the Tigerpaw Database Utility, which is configured specifically to work with the accompanying version of Tigerpaw. Instructions to use the Database Utility are the same whether you use the full or Express version of SQL Server.

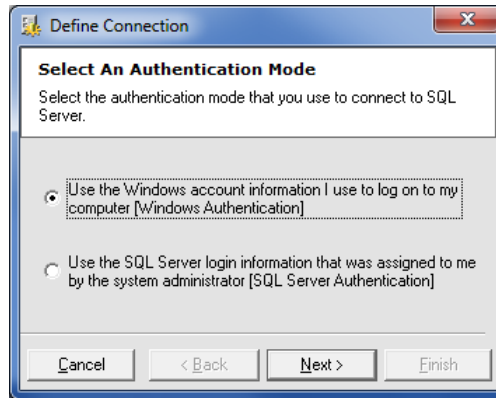
The Database Utility application consists of the following individual utilities:

Utility	Notes
Create Database	Create a database using the Tigerpaw database template files that were auto installed with Tigerpaw. NOTE: You <u>must</u> use this utility to create databases that Tigerpaw can access.
Attach Database	Attach a Tigerpaw database to SQL Server. Tigerpaw can only access databases that have been attached.
Detach Database	Detach a Tigerpaw database from SQL Server. You must detach a database before moving or deleting it.
Back Up Database	Back up a Tigerpaw database. We recommend you perform a daily backup.
Restore Database	Restore a Tigerpaw database that was previously backed up.
Shrink Log File	Shrink the log file of a Tigerpaw database to recover free space on your server.
Clean Up Database	Clean up inventory-related data in a Tigerpaw database.
Clear Lock	Clear a lock that is preventing users from accessing a Tigerpaw database.
Upload Database	Upload a Tigerpaw database to Tigerpaw Software for research and testing purposes.
Run SQL Script	Run a SQL script on a Tigerpaw database to correct bad data or mass update a large number of records.

Open the Tigerpaw Database Utility

Follow these steps to open the Tigerpaw Database Utility:

1. Log into the server.
2. On the Windows desktop, select Start > All Programs > Tigerpaw > Tigerpaw Database Utility. The Define Connection window displays.



For detailed information about authentication modes, consult your SQL Server documentation.

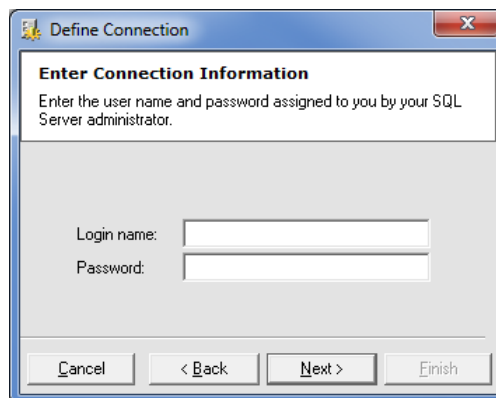
3. Select an authentication mode.

If the SQL Server and client machines running Tigerpaw are members of a Windows domain, retain the default setting. To use Windows authentication, your SQL Server must be configured for Windows or mixed-mode connections.

If you do not want to use Windows authentication or the SQL Server and client machines are not members of a Windows domain, select the option to use the SQL Server login information.

NOTE: If you selected Windows authentication, skip to step 6. If you selected SQL Server authentication, the Enter Connection Information page displays, and you must complete steps 4-5.

4. Click Next. The Enter Connection Information page displays.



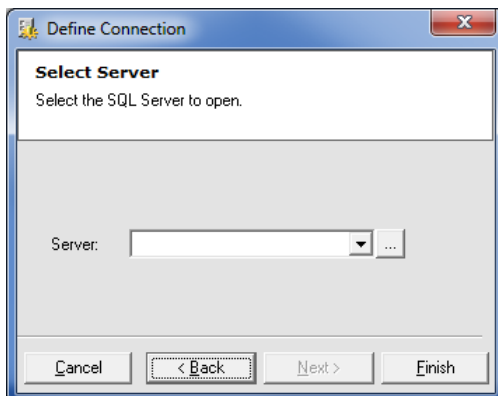
Fields on this page are case sensitive.

5. Enter a valid **Login name** and **Password** for accessing SQL Server.

If you installed the version of SQL Server Express provided by Tigerpaw and you did not modify the default login information, enter **sa** for the **Login name** and **Tigerpaw!1** for the **Password**.

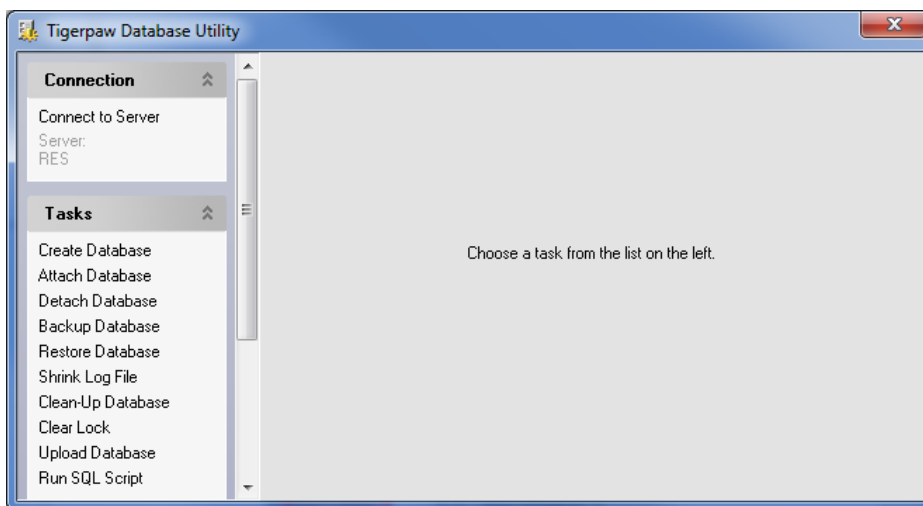
NOTE: We recommend you change the default password.

6. Click Next. The Select Server page displays.



Select the server that hosts your SQL Server application.

7. In the **Server** field, enter the server name or click to select the server from a dropdown list. If the server is not listed, click (Specify server) to enter the server name or IP address. Click OK to save the entered value to the Define Connection window.
8. Click Finish to display the Tigerpaw Database Utility dialog box. The selected server name displays in the Connection section at the top of the navigation pane.



The right pane is blank until you click a task in the left pane.


Create a Tigerpaw Database

Your Tigerpaw business data is stored in a SQL Server database. Every SQL Server database consists of two physical files. The integrity of your database is tied to both files, and both files are required.

Physical Database Files	Description
Master Data File (.MDF)	Actual raw data stored in a series of tables within the SQL database.
Log Data File (.LDF)	Details of all modifications done on your database and details of the transactions that created, modified or deleted the data.

The Create Database utility creates both files. The database must be created on the server where SQL Server resides, and you must have administrator rights on the server to create the database. In addition, the database must reside on a network drive accessible to all workstations.

This utility creates a database using the Tigerpaw database template and then automatically attaches the database to SQL Server, eliminating the need to run the Attach Database utility.

 To minimize issues with SQL Server, avoid spaces and special characters in the database location, .MDF filename and database name. Use only letters, numbers, dashes and underscores.

In addition, place the database on a logical drive, preferably a root drive, such as *C:\Data* on the database server. Do not use a mapped drive.

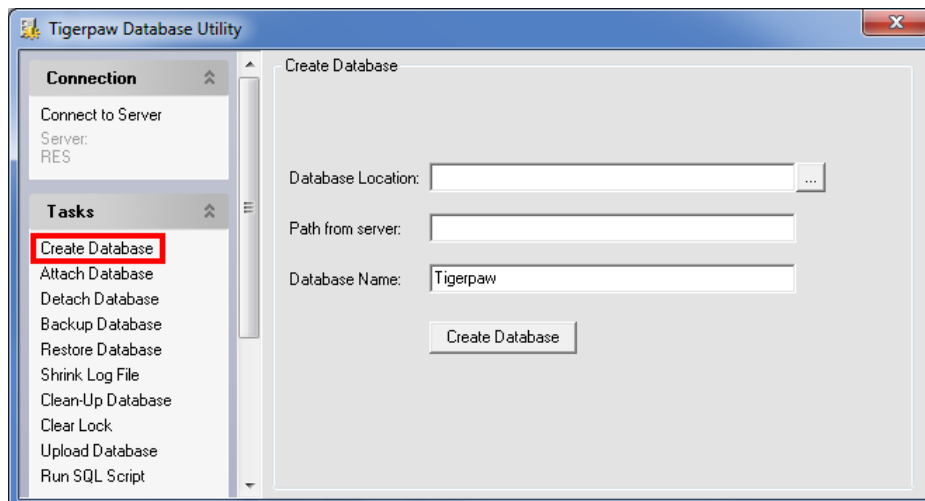


Training Video: Creating a New Database:


<http://support.tigerpawsoftware.com/videos/v10%20Training/FLASH/createnewdatabase/createnewdatabase/createnewdatabase.html>

Follow these steps to create a Tigerpaw database:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. From the Tasks list in the left pane, click Create Database to display the Create Database fields.



The right pane is blank until you click a task in the left pane.

3. Click  (Specify file) beside **Database location**. The Create Database window displays.
4. Navigate to the network folder your company uses to save the physical database files.

The .MDF filename defaults as *Tigerpaw.mdf*, but you can modify the filename or enter a different filename to uniquely identify the database.

5. Click Save. On the Tigerpaw Database Utility window, the **Database location** and **Path from server** fields are now populated. If the **Path from server** value does not match the **Database location path**, modify the **Path from server** value.

NOTE: The **Database location** and **Path from server** fields must specify a local attached drive, not a mapped network drive or remote server.

6. The **Database Name** defaults to 'Tigerpaw'. This value displays when you select a database in Tigerpaw, the Import Wizard, or the Database Utility. If you modify the database name, use only letters, numbers, dashes and underscores. Avoid spaces or special characters.
7. Click Create Database. A message confirms the database was created.

If a message indicates the database was not created, verify you have write permissions to the selected **Database location** and repeat the process.

8. Click  to close the Tigerpaw Database Utility.

Attach an Existing Tigerpaw Database to your SQL Server

The Tigerpaw application can access only databases that are attached to your SQL Server. When you use the Tigerpaw Database Utility to create a database, the utility automatically attaches the database to your SQL Server. However, you must manually attach pre-existing databases such as the Sample database included with the Tigerpaw installation files, or databases you created and then detached, for example to move the database from one server to another.

The Sample database is prepopulated with sample data, enabling you to immediately begin training on Tigerpaw. The default location of this database depends on your operating system:

- For 64-bit machines: *C:\Program Files (x86)\Tigerpaw Software\Tigerpaw\Sample*
- For all other machines: *C:\Program Files\Tigerpaw Software\Tigerpaw\Sample*

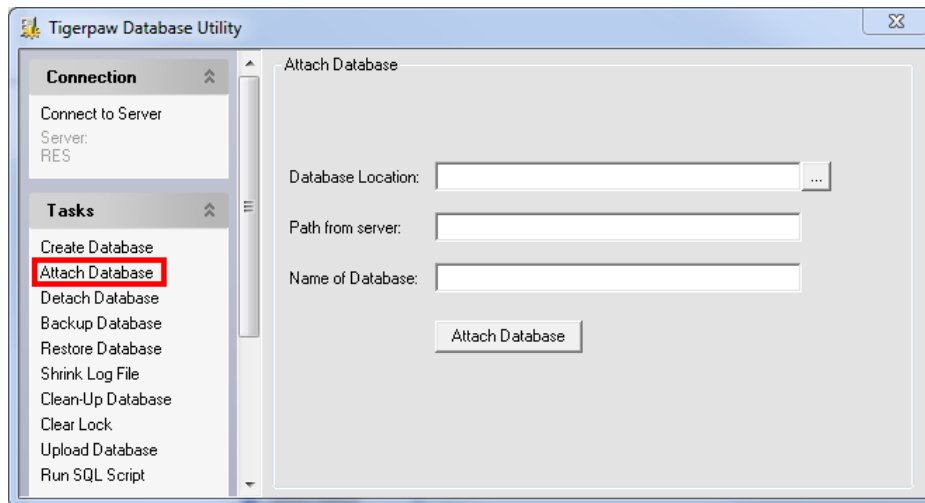


Training Video: Attaching a Sample Database:

http://support.tigerpawsoftware.com/videos/v10%20Training/FLASH/AttachSampleDatabase_train.html

Follow these steps to attach an existing Tigerpaw database to SQL Server:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. From the Tasks list in the left pane, click Attach Database to display the Attach Database fields.




The right pane is blank until you click a task in the left pane.

3. Click (Select file) beside **Database Location**. The Choose Database window displays.
4. Navigate to the *Sample* folder and select *Sample.mdf*, which is the physical master data file (.MDF) for the database.
5. Click Open. On the Tigerpaw Database Utility window, the **Database location** and **Path from server** fields are now populated.
6. In the **Database Name** field, enter the value you want to appear when users select the database in Tigerpaw, the Import Wizard and the Database Utility. Enter only letters, numbers, dashes and underscores. Avoid spaces or special characters.



If the database was previously attached to your SQL Server, you must attach it using the exact same name.

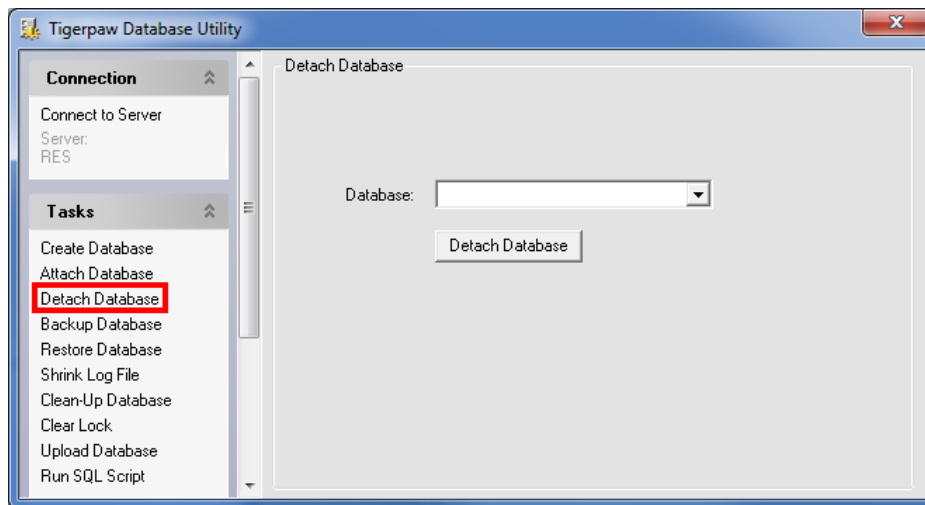
7. Click Attach Database. A message confirms the database was attached.
8. Click  to close the Tigerpaw Database Utility.

Detach a Tigerpaw Database from your SQL Server



If you want to delete a test database or move a database to a different server, you must first detach the database from SQL Server. You can also detach a database to prevent users from accessing it, since detached databases are unavailable to Tigerpaw. The user detaching a database must have system administrator rights on the server where the database resides.

Follow these steps to detach a Tigerpaw database from SQL Server:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. From the Tasks list in the left pane, click Detach Database to display the Detach Database fields.




The right pane is blank until you click a task in the left pane.

3. In the **Database** field, click  to select the database from a dropdown list.
4. Click Detach Database. A message confirms the database was detached.
5. Click  to close the Tigerpaw Database Utility.

Back up a Tigerpaw Database

Creating daily backups of your database is essential to ensuring your business runs smoothly after a catastrophic data loss due to hardware failure, inadvertently importing corrupt data or a natural event such as fire, flood, severe weather or lightning strike. It might not happen tomorrow or next year, but one day **it will happen to you**. Without a current backup of your database, you could lose a significant amount of business data that is not easily replaced.

 Backup equipment is inexpensive, and backing up data is easy to do. A good backup procedure is an inexpensive insurance policy against losing your business operations data.

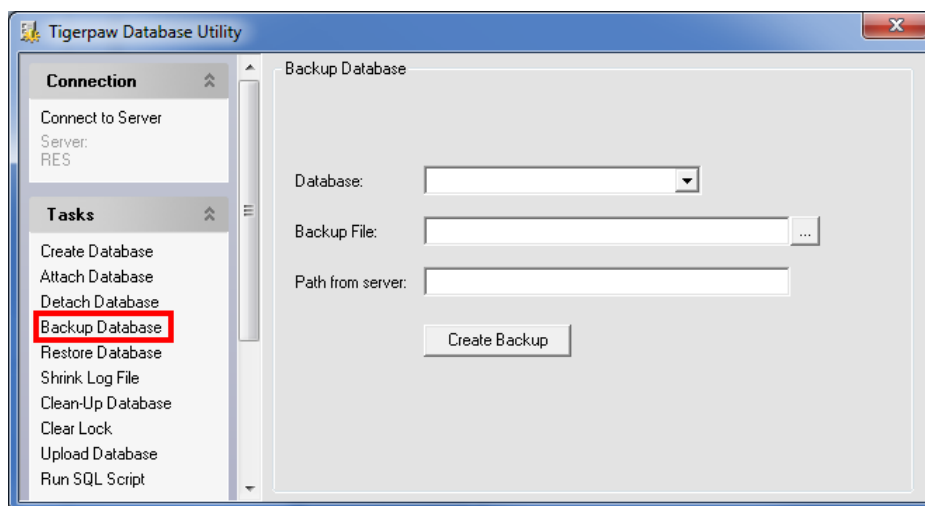
Keep these guidelines in mind as you establish your backup procedures:

- We strongly recommend you back up your database once a day.
- Perform the backup outside of business hours to avoid impacting normal business operations.
- Keep at least one week of database backups. Each time you create a new backup, delete the oldest backup file. This method saves memory resources by keeping only the newest copies on hand.
- Create at least two instances of each backup, with at least one copy saved to an off-site location to protect against theft and natural disasters.
- The user performing the backup must have system administrator rights on the server where the database resides.


Tigerpaw data is stored in a Microsoft SQL Server database. When you back up a database, SQL Server runs certain built-in procedures to preserve transactional information and data. Unlike other file types, you cannot simply copy the SQL Server database files to create a backup. Backing up a SQL Server database requires use of a SQL Server management tool such as the Tigerpaw Database Utility or SQL Server Management Studio. Restoring a SQL Server database also requires one of these tools.



Follow these steps to back up a Tigerpaw database:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. From the Tasks list in the left pane, click Backup Database to display the Backup Database fields.



The right pane is blank until you click a task in the left pane.

3. In the **Database** field, click  to select the database from a dropdown list.

4. Click  (Select file) beside **Backup file**. The Create Backup of Database window displays.
5. Navigate to the network folder your company uses to save backup files.
The backup (BAK) filename defaults as *TigerpawBackup.bak*. Add the backup date or enter a different filename to uniquely identify the backup file and backup date.
6. Click Save. On the Tigerpaw Database Utility window, the **Backup file** and **Path from server** fields are now populated.
7. Click Create Backup. A message confirms the database was backed up.
8. Click  to close the Tigerpaw Database Utility.

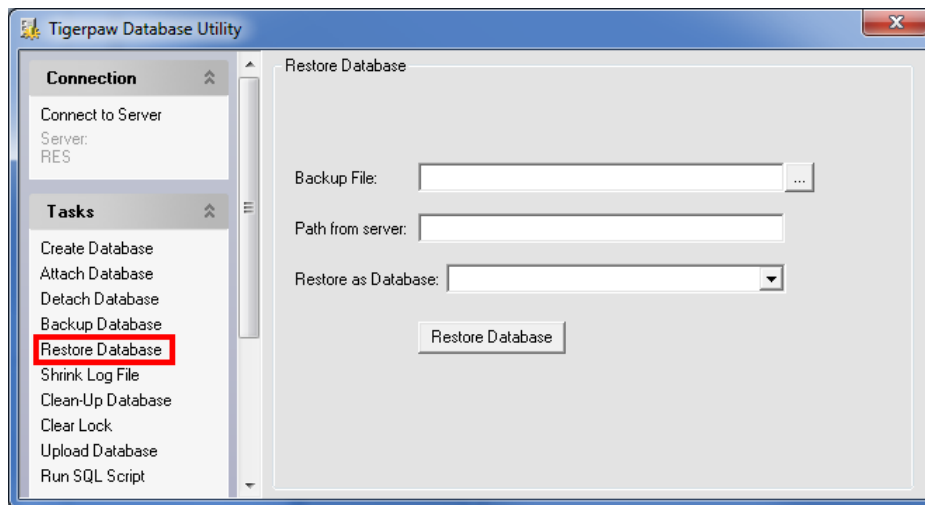
Restore a Tigerpaw Backup Database

When you back up database, SQL Server runs certain built-in procedures to preserve specific transactional information and data. Restoring a backup file to an operational database requires use of a SQL Server management tool such as the Tigerpaw Database Utility or SQL Server Management Studio. When you restore a backup using the Tigerpaw Database Utility, the utility automatically attaches the database to SQL Server.



NOTE: The Tigerpaw Database Utility requires you to restore a database to the original database name. If you have a full version of SQL Server and you want to restore the database to a different name, you must use the tools included with SQL Server.

Follow these steps to restore a Tigerpaw database:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. From the Tasks list in the left pane, click Restore Database to display the Restore Database fields.




The right pane is blank until you click a task in the left pane.

3. Click  (Select file) beside **Backup File**. The Choose Backup File window displays.
4. Navigate to the network folder your company uses to save backup files.
5. Select the backup file you want to restore. Backup files display with a .BAK extension after the file name.
6. Click Open. On the Tigerpaw Database Utility window, the **Backup file** and **Path from server** fields are now populated.
7. In the **Restore as database** field, click  to select the database from a dropdown list.



You must restore the backup database to the original database name. The database utility prevents you from "renaming" the database being restored.

8. Click Restore Database. A message confirms the database was restored.
9. Click  to close the Tigerpaw Database Utility.

Shrink the Log File of your Tigerpaw Database

Every SQL Server database consists of two physical files. The integrity of your database is tied to both files, and both files are required.

Physical Database Files	Description
Master Data File (.MDF)	Actual raw data stored in a series of tables within the SQL database.
Log Data File (.LDF)	<p>Details of all modifications done on your database and details of the transactions that created, modified or deleted the data.</p> <p>The log data file is logically divided into smaller segments called virtual log files. Any modifications made by the SQL Server to the size of the log data file, such as truncating or growing the transaction log files, are performed in units of virtual log files.</p> <p>Your database administrator can configure the log data file to expand to use all available disk space only the specified space.</p>

Over time, the log data file can become quite large. If the log file consumes enough disk space, you may see reduced performance in your database. The Shrink Log File utility removes excess free memory space in the LDF file. The more free space you reclaim, the more the log file shrinks.

NOTE: This process may take several minutes or several hours, depending on the size of your database.



This utility modifies your database.

Before you shrink the log file, you must create a backup copy of your database to ensure you can restore your database if necessary.

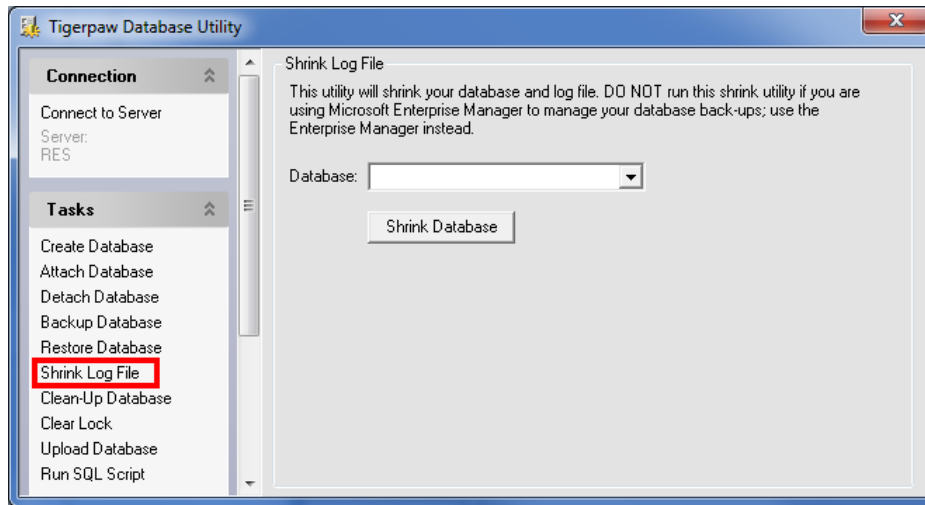


Do not run the Shrink Log File utility if you administer your SQL databases with Microsoft Enterprise Manager or SQL Server Management Studio. These tools provide enhanced features for shrinking database log files. We recommend you shrink your database log files with the tool you use to administer SQL database.


Follow these steps to shrink the size of a database log data file:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. Run the Backup Database utility. Refer to the [Backing up a Tigerpaw Database](#) topic (page 11) for detailed instructions.

- From the Tasks list in the left pane, click Shrink Log File to display the Shrink Log File fields.



The right pane is blank until you click a task in the left pane.

- In the **Database** field, click  to select the database from a dropdown list.
- Click Shrink Database to start the process.
- Click Yes on the message indicating this process may take several minutes or several hours depending on your database size.

When the process is completed, a message confirms the log data file has been shrunk.

- Click  to close the Tigerpaw Database Utility.

Clean up Inventory Data in a Tigerpaw SQL Database

The Clean Up Database utility corrects database inconsistencies related to inventory, such as inventory count problems that may cause invalid operations within Tigerpaw. Refer to the [Database Maintenance Performed by the Clean Up Utility](#) topic (page 17) for a detailed list of the processes completed by this utility.



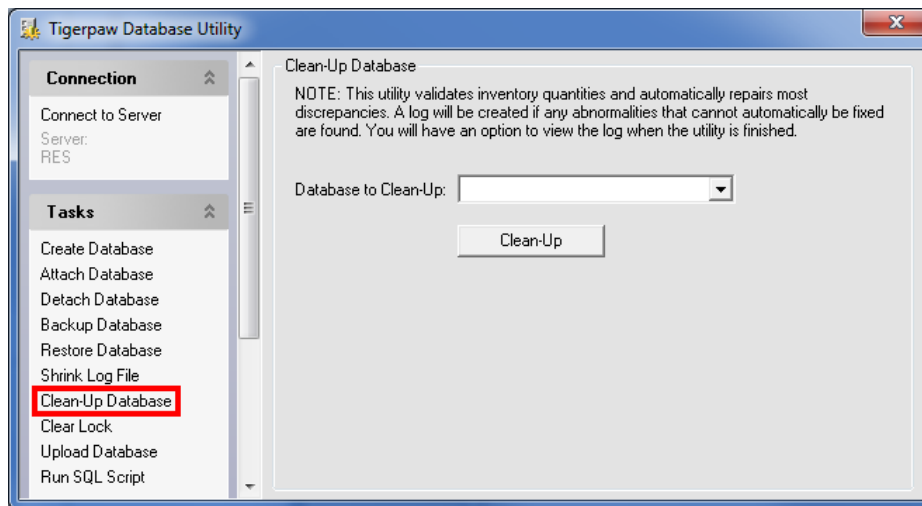
This utility modifies your database.

Before cleaning up the database, you must create a backup copy to ensure you can restore your database if necessary.


All users must be out of the database when you run this utility. The user running the utility must have system administrator rights on the server where the database resides and be a system administrator on the Tigerpaw database.

Follow these steps to run the database cleanup utility on your Tigerpaw database:

1. Ensure all users are out of the database.
2. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
3. Run the Backup Database utility. Refer to the [Backing up a Tigerpaw Database](#) topic (page 11) for detailed instructions.
4. From the Tasks list in the left pane, click Clean Up Database to display the Clean Up Database fields.



The right pane is blank until you click a task in the left pane.

5. In the **Database to clean up** field, click  to select the database from a dropdown list.
6. Click Clean Up to start the process.
7. At the prompt, enter or select your user name. You must be a system administrator in Tigerpaw.
8. Click OK.
9. Click Yes on the message verifying that you have backed up your database and no one else is logged into the database.

An error log is created if the utility encounters abnormalities that cannot be fixed automatically. The log lists the errors and possible solutions, and indicates you must fix the errors and run the Clean Up Database utility again.

10. If necessary, correct issues noted in the error log and run the utility again. A database cleanup is successful when it runs without generating an error log.

11. Click  to close the Tigerpaw Database Utility.

Database Maintenance Performed by the Inventory Clean Up Utility

The Clean Up Database utility performs the following processes on your database:

Database Table	Database Maintenance Step
tblPriceBook	Strip double quotes from ItemID and ItemDescription fields. NOTE: If double quotes are stripped from the ItemID field, this utility updates all other tables that contain this ItemID value.
tblPriceBook	Set 'N' (non-stock) to 'S' (special) in the item Type field.
tblQuoteDetail tblQuoteAssemblyDetail tblSOPartsUsed tblSOPartsUsedAssembly Detail	Set detail Type to match the Type in tblPriceBook.
tblInvoices	If WorkOrderNumber contains a value but QuoteNumber is blank, enter the QuoteNumber tied to the WorkOrderNumber on tblQuotes.
tblInvoiceDetail tblQuoteDetail tblQuoteAssemblyDetail tblSOPartsUsed tblSOPartsUsedAssembly Detail	Format all quantity fields to 2 decimal places.
tblSOPartsUsed tblSOPartsUsedAssembly Detail	If the service order is invoiced in full, set InvoiceNumber to match the InvoiceNumber in tblServiceOrders for the transaction. If the service order is not invoiced in full, clear the InvoiceNumber field.
tblSOPartsUsed tblSOPartsUsedAssembly Detail tblPriceBook	If the related service order Status is 'Void' on tblServiceOrders, clear the QuantityFilled field and enter a quantity to match the tblPriceBook QuantityOnHand field.
tblServiceOrders	Set PostedStatus to match the Status on the associated tblInvoices record.
tblQuoteDetail tblQuoteAssemblyDetail	If the quote is tied to a posted invoice, set the quote detail PostedDate to match the tblInvoices StatusDate.
tblInvoiceDetail tblInvoiceAssemblyDetail	If the invoice is posted, set the detail QuantityFilled and QuantityPosted values to match the Quantity.

tblQuoteDetail tblQuoteAssemblyDetail	If PostedDate is populated on tblQuotes for the quote, set the quote detail QuantityPosted to match the QuantityFilled.
tblSOPartsUsed tblSOPartsUsedAssemblyDetail	If the service order is invoiced in full and the invoice is posted, set QuantityPosted to match QuantityFilled.
tblSOPartsUsed	If the service order is partially invoiced, set QuantityPosted to match the QuantityPosted on tblInvoiceDetail. This function makes a recordset of posted tblInvoiceDetail items for service orders, groups the recordset by FKSourceDetail and calculates totals by the quantity posted. The record set is then traversed and the quantity entered in the corresponding QuantityPosted field on tblSOPartsUsed.
tblSOPartsUsedAssemblyDetail	If the service order is partially invoiced, set QuantityPosted to match QuantityPosted on tblInvoiceAssemblyDetail. This function makes a record set of posted tblInvoiceAssemblyDetail items for service orders, groups the record set by FKSourceAssemblyDetail and calculates totals by the quantity posted. The record set is then traversed and the quantity entered in the corresponding QuantityPosted field on tblSOPartsUsedAssemblyDetail.
tblDebitMemoDetail tblDebitMemoAssemblyDetail	If the debit memo is posted, set QuantityPosted to match QuantityFilled.
tblSOPartsUsed tblSOPartsUsedAssemblyDetail tblQuoteDetail tblQuoteAssemblyDetail tblDebitMemoDetail tblDebitMemoAssemblyDetail	If QuantityFilled is greater than Quantity, change QuantityFilled to match Quantity.
tblSOPartsUsed tblSOPartsUsedAssemblyDetail tblQuoteDetail tblQuoteAssemblyDetail	If a material or finished good item was fractionally filled, set QuantityFilled to the integer value.
tblServiceOrders	If StagingLocation is blank on tblServiceOrders but Location is populated in tblInvoices, set the service order StagingLocation to match the invoice Location. If no location can be found, unfill all items for the order and release serial numbers.
tblQuotes	If StagingLocation is blank on tblQuotes but Location is populated in tblInvoices, set the quote StagingLocation to match the invoice Location. If no location can be found, unfill all items for the quote and release serial numbers.

tblSOPartsUsed tblSOPartsUsedAssembly Detail	If StagingLocation is blank on the associated tblServiceOrders record, set QuantityFilled to zero.
tblQuoteDetail tblQuoteAssemblyDetail	If StagingLocation is blank on the associated tblQuotes record, set QuantityFilled to zero.
tblSOPartsUsed tblSOPartsUsedAssembly Detail tblQuoteDetail tblQuoteAssemblyDetail tblDebitMemoDetail tblDebitMemoAssemblyDe tail	If the item does not exist in the price book, set QuantityFilled to zero.
tblAssemblyDetail	Delete tblAssemblyDetail records with a blank or missing AssemblyHeaderItemID.
tblExchange	If the related service order invoice Status is 'Posted', set the Status to 'Posted'.
tblSerialNumbers	<p>The following processes are applied:</p> <ul style="list-style-type: none"> ● If a numeric field value is Null and the default should be zero, set the value to zero. ● Delete the record if the Status is other than 'Sold' and one of the following conditions applies: <ul style="list-style-type: none"> ■ The ItemID does not exist in the price book. ■ Serialized is set to 'No' on the price book record for the ItemID. ● Release reserved serial numbers if QuantityFilled is zero on the related service order, quote or debit memo. ● If Location is blank on a Reserved or Invoiced serial number, set Location to match the StagingLocation on tblServiceOrders, tblQuotes or tblDebitMemos, using the assigned sequence. ● If Location is blank on an Available serial number, assign the default warehouse location.
tblInvoiceDetail tblInvoiceAssemblyDetail	Set Serialized to 'No' if Serialized is set to 'No' on the price book record for the ItemID.

tblLocator	<p>The following processes are applied:</p> <ul style="list-style-type: none"> ● Recalculate QuantityReserved to match the sum of QuantityFilled in order detail records, including tblDebitMemoAssemblyDetail, tblDebitMemoDetail, tblQuickSaleAssemblyDetail, tblQuickSaleDetail, tblQuoteAssemblyDetail, tblQuoteDetail, tblRMADetail, tblSOPartsUsedAssemblyDetail and tblSOPartsUsed. ● Set QuantityReserved to zero for all items; sum all filled and non-posted order detail items by location and ItemID; and use the totals to update QuantityReserved in tblLocator. If the new QuantityReserved is greater than Quantity, write a log record. ● If no tblLocator record exists for a serialized price book record with quantity on hand, write a log record. ● If the tblLocator Quantity is different from the tblPriceBook QuantityOnHand for a serialized item, write a log record. ● If the If the tblLocator QuantityReserved is different from the tblPriceBook QuantityReserved for a serialized item, write a log record. <p>At this point, the tblLocator quantities should match the actual values. None of these changes affect inventory valuation, since no changes are made to on hand quantities. The price book and costing are now updated for the current counts.</p>
tblLocator	<p>The following processes are applied:</p> <ul style="list-style-type: none"> ● For each item in tblLocator, update QuantityOnOrder to match the sum of QuantityOrdered on tblIPODetail minus the sum of QuantityReceived on tblPORReceipts. ● For each item in tblLocator, update QuantityReserved to match the sum of unfilled quantities on non-voided service orders and work orders.
tblPriceBook	<p>Update the QuantityOnHand, QuantityReserved and QuantityOnOrder fields to match values in the corresponding tblLocator fields.</p>
tblSOItemsServiced	<p>The following processes are applied if the RMA number field is populated but no corresponding detail record exists for the RMA:</p> <ul style="list-style-type: none"> ● Set FKRMA to zero. ● Set FKRMADetail to zero. ● Set RMA number to Null.

tblFIFoLIFO	<p>The following processes are applied:</p> <ul style="list-style-type: none"> ● For each tblPriceBook item with quantity on hand and no record in tblFIFoLIFO, create a record. Set ReceiptDate to the current date. Set ReceiptQuantity to the QuantityOnHand in tblPriceBook. The user must specify whether to enter the tblPriceBook AverageCost or BaseCost in ReceiptUnitCost. ● Delete tblFIFoLIFO records with no corresponding price book record. ● If the tblPriceBook record has a quantity of zero, delete all tblFIFoLIFO records for the same item. ● For each item, compare the sum of ReceiptQuantity values to the QuantityOnHand value in tblPriceBook. If quantities match, no change is made. If quantities don't match, adjust the tblFIFoLIFO record: <ul style="list-style-type: none"> ■ To reduce quantity: Make a record set of all pools (i.e. collection of tblFIFoLIFO records) for an item and sort by receipt date (ascending if FIFO and descending if LIFO). If the total pool quantity is greater than the price book quantity, reduce the tblPriceBook quantity to match the price book. <p>If the pool quantity changes to zero, remove the pool. If the pool quantity would go negative, remove the pool and update the next pool until the necessary quantity adjustment is made.</p> <ul style="list-style-type: none"> ■ To increase quantity: Enter the additional quantity on the first tblFIFoLIFO record in the record set. If no record exists, create one. Set ReceiptUnitCost to the tblPriceBook BaseCost, AverageCost or LatestCost based on the first record found for the item in the record set.
tblFIFoLIFO	<p>If Cost method on the Configure Accounting window is set to Average or Standard, delete all records.</p>
tblPriceBook	<p>If Cost method on the Configure Accounting window is set to FIFO or LIFO, complete the following steps:</p> <ul style="list-style-type: none"> ● Set AverageCost and TotalCost to zero. ● Set TotalCost to match the sum of ReceiptUnitCost values on tblFIFoLIFO. ● Set AverageCost to the calculated average cost based on the new TotalCost value.
tblPriceBook	<p>If Cost method on the Configure Accounting window is set to Average or Standard, complete the following steps:</p> <ul style="list-style-type: none"> ● If QuantityOnHand is zero, set AverageCost and TotalCost to zero. ● If QuantityOnHand is greater than zero and the costing method is Average, set TotalCost to match AverageCost * QuantityOnHand. ● If QuantityOnHand is greater than zero and no average cost exists, set TotalCost to match LatestCost * QuantityOnHand. If LatestCost is zero, use BaseCost. ● Set AverageCost to the calculated average cost based on the new TotalCost value.

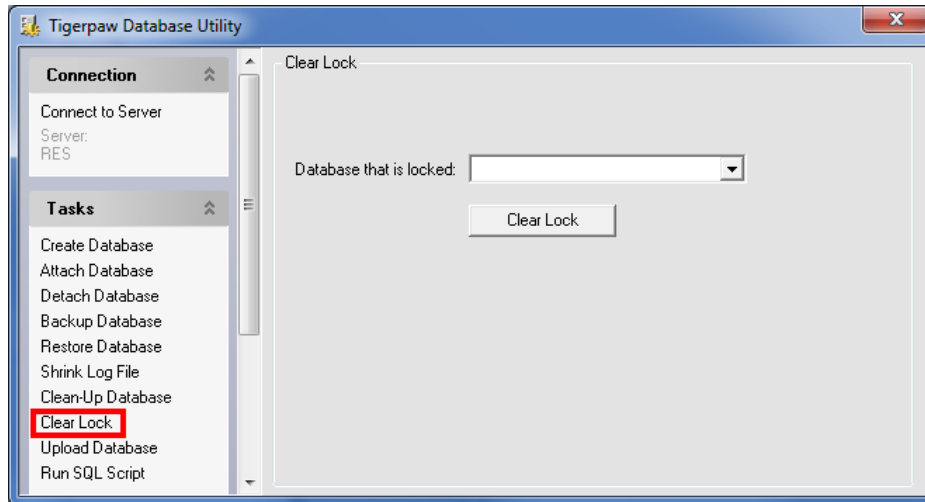
Clear a Lock on a Tigerpaw Database

If you are logged into Tigerpaw when your workstation crashes or you power off the workstation without logging out of Tigerpaw, your Windows system recognizes that the Tigerpaw application session is terminated, but SQL server thinks you are still connected to the database. Typically when you log back into Tigerpaw, a message indicates you are still connected and verifies if you want to end ("break") the existing connection. When you click Yes, you are able to log into Tigerpaw.



Sometimes, however, the loss of connection ties up your database and no users can log in. In this situation, you must manually break the erroneous connection by clearing the lock. The Clear Lock utility removes the lingering connection and allows users to re-access the database. Typically this process takes just a few seconds. The user clearing the database lock must have system administrator rights on the server where the database resides.

Follow these steps to clear a lock on your Tigerpaw database:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. From the Tasks list in the left pane, click Clear Lock to display the Clear Lock fields.



The right pane is blank until you click a task in the left pane.

3. In the **Database that is locked** field, click  to select the database from a dropdown list.
4. Click Clear Lock. A message confirms the exclusive lock on the database has been cleared. Users can now log back into Tigerpaw.
5. Click  to close the Tigerpaw Database Utility.

Upload a Copy of your Database to Tigerpaw Software

If you have a database issue that cannot be easily resolved over the phone, the Tigerpaw rep may ask you to send us a copy of your database for research and testing purposes. In this situation, the Tigerpaw rep may ask you to upload a copy of your database to Tigerpaw Software. The user uploading the database must have system administrator rights on the server where the database resides.



Before uploading a database, you must have a file ID provided by Tigerpaw Software. Without a file ID, your database cannot be identified or matched to the related service order and will most likely be deleted upon receipt at Tigerpaw Software.

When you use the Upload Database utility, the following processes occur:

- The utility creates a copy of your database, leaving your live data intact.
- The copy is compressed to achieve a more manageable file size.
- The compressed file is encrypted for security purposes.
- The file is auto uploaded to our secure FTP site.
- When the upload is complete, the Tigerpaw Help Desk is auto notified that the database is available.

The Tigerpaw rep uses the file ID to match your database to the appropriate service order. The rep removes the database from our FTP site and downloads it to a secure server, where it is attached to a SQL Server for research and testing purposes. Downloading and decompressing your database requires a special internal password that is available to a limited number of Tigerpaw employees. These procedures ensure your data is always secure.

When the research and testing is completed, the Tigerpaw rep will advise you of the outcome and then permanently delete your database so your data is irretrievable. This step ensures your data is not compromised and keeps available disk space on our servers at an optimal level.

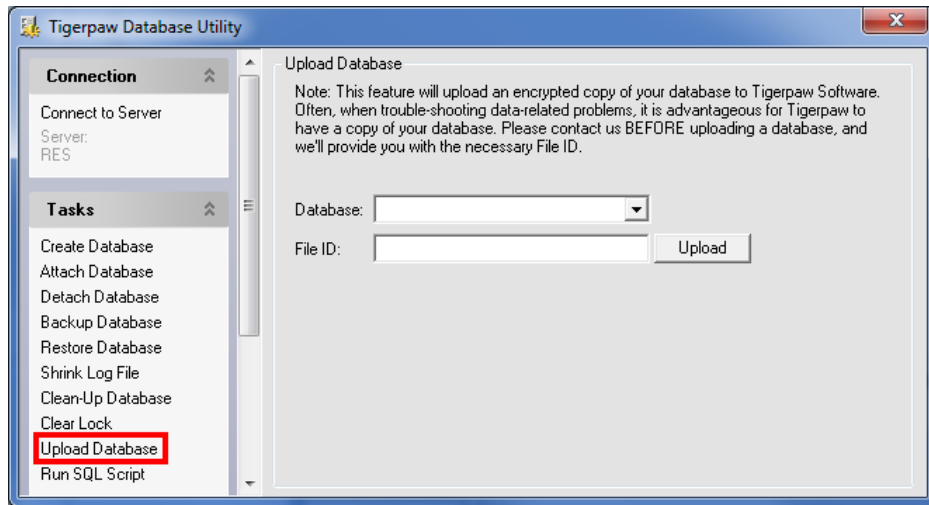


The database upload may take several hours to run, based on the size of your database and your internet connection speed.



Follow these steps to upload a copy of your database directly to Tigerpaw Software:

1. Log onto your database server.
2. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.

3. From the Tasks list in the left pane, click Upload Database to display the Upload Database fields.



The right pane is blank until you click a task in the left pane.

4. In the **Database** field, click  to select the database from a dropdown list.
5. In **File ID**, enter the identification code provided by Tigerpaw Software.
If you miskey the file ID, a message indicates you must enter a valid file ID. If you receive this message after entering the exact code provided by Tigerpaw Software, contact our Help Desk to verify the code.
6. Click Upload. While the upload is in progress, status messages indicate the current upload stage. When the upload is complete, a message indicates the upload was successful.
NOTE: If any error messages display, contact the Tigerpaw Help Desk for further assistance.
7. Click  to close the Tigerpaw Database Utility.

Run a SQL Script on your Tigerpaw Database

In certain cases, you may need to correct bad data or modify large amounts of data at one time, rather than touch each record individually. This process can be accomplished through a SQL script. A SQL script is a file written in structured query language (SQL) that contains programming statements written to add, delete or modify data in a database. The user running the SQL script must have system administrator rights on the server where the database resides.

SQL scripts change data in your database and should be run only if provided or approved by Tigerpaw Software. If you run a SQL script not provided or approved by Tigerpaw Software, the script may adversely affect your data or, in extreme cases, could render your database inoperable. Depending on the circumstances, reviewing and assessing the impact on your database may fall outside the scope of your software support contract, resulting in a fee to repair your database.



SQL scripts modify your database.

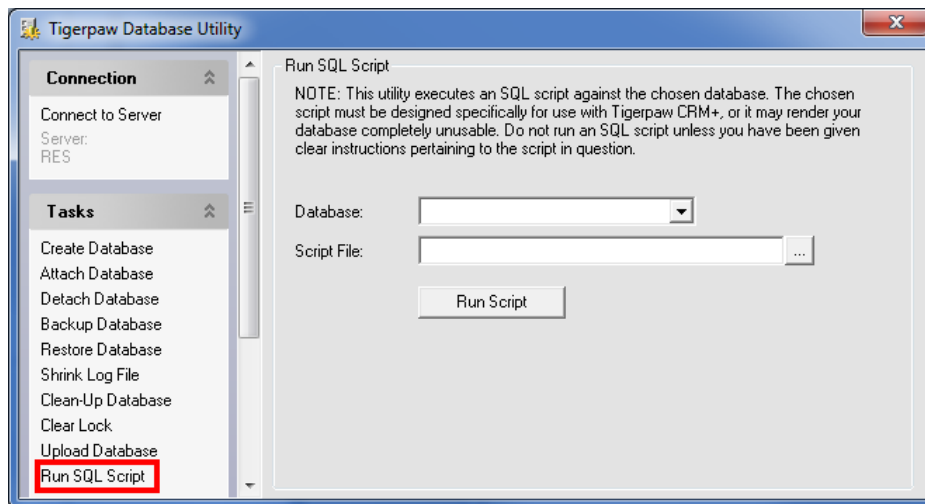
You should exercise great care in running a SQL script on your database, only after consulting with Tigerpaw Software and obtaining the script from Tigerpaw.

Before you run a SQL script, you must create a backup copy of your database to ensure you can restore your database if necessary.



SQL scripting is generally considered a customization to your database. Therefore, SQL scripts provided by Tigerpaw are typically fee-based.

Follow these steps to run a SQL script on your Tigerpaw database:

1. Open the Tigerpaw Database Utility. Refer to the [Opening the Database Utility](#) topic (page 4) for detailed instructions.
2. Run the Backup Database utility. Refer to the [Backing up a Tigerpaw Database](#) topic (page 11) for detailed instructions.
3. From the Tasks list in the left pane, click Run SQL Script to display the Run SQL Script fields.



The right pane is blank until you click a task in the left pane.

4. In the **Database** field, click  to select the database from a dropdown list.
5. Click  (Select file) beside **Script file**. The Choose SQL Script window displays.

6. Navigate to the network folder containing the script file. SQL script files display with a .SQL extension after the file name.
7. Click Run Script. While the script is running, status messages indicate the current process. When the script is finished, a message indicates the script was successfully executed.

NOTE: If any error messages display, contact the Tigerpaw Help Desk for further assistance.

8. Click  to close the Tigerpaw Database Utility.