phpMSAdmin

Written By: Adam J. Voigt (avoigt@phpmsadmin.org)

Thank you for downloading phpMSAdmin. This software is designed to be a capable replacement for the proprietary Microsoft SQL Enterprise Manager and the newer Microsoft SQL Server Management Studio. By using phpMSAdmin you can manage your SQL server from any web browser, hosting the software on any OS capable of running the free and open-source PHP development language. Tutorials for installing phpMSAdmin with PHP, FreeTDS (the open-source SQL library) and Apache are available from the phpMSAdmin website (http://www.phpmsadmin.org).

This manual is a work in progress and might not answer every question you may have. If you have a question which is not answered, feel free to visit the forums (http://phpmsadmin.sourceforge.net/forum/).

Current Limitations

User creation is at a somewhat stable state however user modification is not currently implemented.

Changing Options

All user-accessible configuration is done via the “config.php” file in the “inc” directory. From this point forward in the documentation, you should assume that when the “configuration file” is referenced, it is referring to the above mentioned file.

Connecting To A Server

The connection procedure is rather automatic. If you are hosting the software on Linux or Unix, the software will attempt to find your FreeTDS configuration file via the “freetdspath” option in the configuration file. If it is unable to find your FreeTDS configuration file and you have not defined any manual server addresses via the “connections” setting in the configuration file, then the software will switch in to manual mode where you are given a standard text input box to enter your server address. While in some server configurations this may work, it is highly suggested you correct the FreeTDS configuration file / path. If the software is being hosted on Windows, the software will attempt to locate and parse your Windows data-sources via the “odbcinipath” in the configuration file. If it is unable to locate this file, or if it is empty and you have not defined the “connections” variable, then you will be switched into manual connection specification mode, and be given a standard text input to enter your server address. If you wish to completely bypass all automatic configuration attempts, you may set the variable “detectionoff” to “true” to be always given a manual input field regardless of any FreeTDS, ODBC, or manual connection settings.

Accessing Tables

A quick table list is accessible by clicking on the “+” sign next to the database of your choice in the menu. The menu will refresh with the database you selected, expanding to list it's tables. You can also access the table list by clicking on the database itself to receive a full database summary page including not only the tables associated with that database but also it's functions, stored procedures, and views. If you choose to expand the menu to list the tables, you now have the option of using the context menu (right-click) on the tables themselves. For instance, by right-clicking on a table you have options to browse, select, insert, and manage triggers as well as more destructive options like empty, rename or drop. The destructive operations will prompt you with a warning before the operation will be executed unless you have turned “dangerous_table_action_prompts” off in the configuration file.

Editing Rows

Rows can be edited from either the table browse or table select functions. However, only tables which have an identity column defined can be edited. When you select or browse on a table with an identity column, in the rows returned, the identity columns value will be a link to edit that row. It does not matter if the identity column is the first column returned or not.

Exporting Data

Data can be exported in one of two formats, CSV or SQL. CSV is designed for use when you want to get the data out of a database and import it into Excel or a custom script. SQL is designed for when you wish to export the data for insertion into another database on the same or on a different SQL server.

To export whole databases or single tables in SQL format, you should use the database export feature. This is available by either right-clicking on the database in the menu or by selecting all the tables in database on the database properties page, then selecting “Export – SQL” from the “Mass Operations” panel. When you are on the database export page, you will see a list of tables. You must selected each and every table you want to be included in the export. If you wish all the tables to be included, you can save some time by clicking on the “All” button at the bottom of the table list. Once you have selected all the tables you wish to be included, you must now set what options you'd like for your export. Under the “Selection” sub-section, you choose what you'd like to save in your export. If you check “Structure” you will get the SQL statements necessary to rebuild your tables. If you check “Data” you will get the SQL statements necessary to re-insert all the data into your tables. The next sub-section is “Preparation” which tells the software how you want to prepare the connection for importing the export file. If you check “Drop Database First”, a database drop statement will be added to the top of the export file. For example, if you do a full export of the “NorthWinds” database as a backup, and then need to roll-back to that backup, the “Drop Database First” option will completely delete your existing (but now bad) database. If you check the “Create Database” option, the statement to actually create your database will be included in the export. If you do not check this box, then your database your importing must have already been created manually. The next option is “Add USE Database Switch”, this simply tells the software to automatically switch SQL server to the correct database before importing, otherwise your tables and data could end up in your user-accounts default database. The final sub-section of export options is “Method”. This describes how you want your data to be handled in the export. If you check “Base-64 Encode Strings” then all your non-numeric values will be Base-64 encoded for easy storage. This is especially handy for fields storing binary data which can, and frequently will, cause problems on import. Note that if you are going to be importing the generated export file via a method other then phpMSAdmin, you will need to make allowances for this option as other methods probably won't make allowances for fields coming into the system Base-64 encoded.

To export data in CSV format, simply select the tables you'd like to export data from on the database properties page, then select “Export – CSV” from the “Mass Operations” menu. On the proceeding page, you will see two options. The first, “Field Delimiter”, specifies how fields are separated in your export, the default is a comma. However, if you are going to be exporting data that already has a comma in it, and your program you're importing into is not smart enough to follow quotes, then you will need to specify an alternate character here. The second option is “Field Quoted With”. This lets you select between single or double quotes around the values in your data.

In either method, once you've selected your export options you will be taken to your export display page where you will see your raw export in a text field. This is mainly for preview purposes, however you could copy and paste from this field into a text file should wish to. An easier method to download the file however, is to click the “Save to File” button which will prompt you for a download location. If you are exporting a CSV file, the default filename will be the database name plus the extension “CSV”. If you are exporting a SQL file, the default file name will be the database name plus the extension “SQL”.