.. index:: config.inc.php

.. \_config:

Configuration

=============

Almost all configurable data is placed in :file:`config.inc.php`. If this file

does not exist, please refer to the :ref:`setup` section to create one. This

file only needs to contain the parameters you want to change from their

corresponding default value in :file:`libraries/config.default.php`.

If a directive is missing from your file, you can just add another line with

the file. This file is for over-writing the defaults; if you wish to use the

default value there's no need to add a line here.

The parameters which relate to design (like colors) are placed in

:file:`themes/themename/layout.inc.php`. You might also want to create

:file:`config.footer.inc.php` and :file:`config.header.inc.php` files to add

your site specific code to be included on start and end of each page.

.. note::

Some distributions (eg. Debian or Ubuntu) store :file:`config.inc.php` in

``/etc/phpmyadmin`` instead of within phpMyAdmin sources.

.. warning::

:term:`Mac` users should note that if you are on a version before

:term:`Mac OS X`, PHP does not seem to

like :term:`Mac` end of lines character (``\r``). So

ensure you choose the option that allows to use the \\*nix end of line

character (``\n``) in your text editor before saving a script you have

modified.

Basic settings

--------------

.. config:option:: $cfg['PmaAbsoluteUri']

:type: string

:default: ``''``

Sets here the complete :term:`URL` (with full path) to your phpMyAdmin

installation's directory. E.g.

``http://www.example.net/path\_to\_your\_phpMyAdmin\_directory/``. Note also

that the :term:`URL` on most of web servers are case–sensitive. Don’t

forget the trailing slash at the end.

Starting with version 2.3.0, it is advisable to try leaving this blank. In

most cases phpMyAdmin automatically detects the proper setting. Users of

port forwarding will need to set :config:option:`$cfg['PmaAbsoluteUri']`

(`more info <https://sourceforge.net/p/phpmyadmin/support-requests/795/>`\_).

A good test is to browse a table, edit a row and save it. There should be

an error message if phpMyAdmin is having trouble auto–detecting the correct

value. If you get an error that this must be set or if the autodetect code

fails to detect your path, please post a bug report on our bug tracker so

we can improve the code.

.. seealso:: :ref:`faq1\_40`

.. config:option:: $cfg['PmaNoRelation\_DisableWarning']

:type: boolean

:default: false

Starting with version 2.3.0 phpMyAdmin offers a lot of features to

work with master / foreign – tables (see :config:option:`$cfg['Servers'][$i]['pmadb']`).

If you tried to set this

up and it does not work for you, have a look on the :guilabel:`Structure` page

of one database where you would like to use it. You will find a link

that will analyze why those features have been disabled.

If you do not want to use those features set this variable to ``true`` to

stop this message from appearing.

.. config:option:: $cfg['SuhosinDisableWarning']

:type: boolean

:default: false

A warning is displayed on the main page if Suhosin is detected.

You can set this parameter to ``true`` to stop this message from appearing.

.. config:option:: $cfg['LoginCookieValidityDisableWarning']

:type: boolean

:default: false

A warning is displayed on the main page if the PHP parameter

session.gc\_maxlifetime is lower than cookie validity configured in phpMyAdmin.

You can set this parameter to ``true`` to stop this message from appearing.

.. config:option:: $cfg['ServerLibraryDifference\_DisableWarning']

:type: boolean

:default: false

A warning is displayed on the main page if there is a difference

between the MySQL library and server version.

You can set this parameter to ``true`` to stop this message from appearing.

.. config:option:: $cfg['ReservedWordDisableWarning']

:type: boolean

:default: false

This warning is displayed on the Structure page of a table if one or more

column names match with words which are MySQL reserved.

If you want to turn off this warning, you can set it to ``true`` and

warning will no longer be displayed.

.. config:option:: $cfg['TranslationWarningThreshold']

:type: integer

:default: 80

Show warning about incomplete translations on certain threshold.

.. config:option:: $cfg['SendErrorReports']

:type: string

:default: ``'ask'``

Sets the default behavior for JavaScript error reporting.

Whenever an error is detected in the JavaScript execution, an error report

may be sent to the phpMyAdmin team if the user agrees.

The default setting of ``'ask'`` will ask the user everytime there is a new

error report. However you can set this parameter to ``'always'`` to send error

reports without asking for confirmation or you can set it to ``'never'`` to

never send error reports.

This directive is available both in the configuration file and in users

preferences. If the person in charge of a multi-user installation prefers

to disable this feature for all users, a value of ``'never'`` should be

set, and the :config:option:`$cfg['UserprefsDisallow']` directive should

contain ``'SendErrorReports'`` in one of its array values.

.. config:option:: $cfg['ConsoleEnterExecutes']

:type: boolean

:default: false

Setting this to ``true`` allows the user to execute queries by pressing Enter

instead of Ctrl+Enter. A new line can be inserted by pressing Shift + Enter.

The behaviour of the console can be temporarily changed using console's

settings interface.

.. config:option:: $cfg['AllowThirdPartyFraming']

:type: boolean

:default: false

Setting this to ``true`` allows phpMyAdmin to be included inside a frame,

and is a potential security hole allowing cross-frame scripting attacks or

clickjacking.

Server connection settings

--------------------------

.. config:option:: $cfg['Servers']

:type: array

:default: one server array with settings listed below

Since version 1.4.2, phpMyAdmin supports the administration of multiple

MySQL servers. Therefore, a :config:option:`$cfg['Servers']`-array has been

added which contains the login information for the different servers. The

first :config:option:`$cfg['Servers'][$i]['host']` contains the hostname of

the first server, the second :config:option:`$cfg['Servers'][$i]['host']`

the hostname of the second server, etc. In

:file:`libraries/config.default.php`, there is only one section for server

definition, however you can put as many as you need in

:file:`config.inc.php`, copy that block or needed parts (you don't have to

define all settings, just those you need to change).

.. note::

The :config:option:`$cfg['Servers']` array starts with

$cfg['Servers'][1]. Do not use $cfg['Servers'][0]. If you want more

than one server, just copy following section (including $i

incrementation) serveral times. There is no need to define full server

array, just define values you need to change.

.. config:option:: $cfg['Servers'][$i]['host']

:type: string

:default: ``'localhost'``

The hostname or :term:`IP` address of your $i-th MySQL-server. E.g.

``localhost``.

Possible values are:

\* hostname, e.g., ``'localhost'`` or ``'mydb.example.org'``

\* IP address, e.g., ``'127.0.0.1'`` or ``'192.168.10.1'``

\* dot - ``'.'``, i.e., use named pipes on windows systems

\* empty - ``''``, disables this server

.. note::

phpMyAdmin supports connecting to MySQL servers reachable via IPv6 only.

To connect to an IPv6 MySQL server, enter its IPv6 address in this field.

.. config:option:: $cfg['Servers'][$i]['port']

:type: string

:default: ``''``

The port-number of your $i-th MySQL-server. Default is 3306 (leave

blank).

.. note::

If you use ``localhost`` as the hostname, MySQL ignores this port number

and connects with the socket, so if you want to connect to a port

different from the default port, use ``127.0.0.1`` or the real hostname

in :config:option:`$cfg['Servers'][$i]['host']`.

.. config:option:: $cfg['Servers'][$i]['socket']

:type: string

:default: ``''``

The path to the socket to use. Leave blank for default. To determine

the correct socket, check your MySQL configuration or, using the

:command:`mysql` command–line client, issue the ``status`` command. Among the

resulting information displayed will be the socket used.

.. config:option:: $cfg['Servers'][$i]['ssl']

:type: boolean

:default: false

Whether to enable SSL for the connection between phpMyAdmin and the MySQL server.

When using the ``'mysql'`` extension,

none of the remaining ``'ssl...'`` configuration options apply.

We strongly recommend the ``'mysqli'`` extension when using this option.

.. config:option:: $cfg['Servers'][$i]['ssl\_key']

:type: string

:default: NULL

Path to the key file when using SSL for connecting to the MySQL server.

For example:

.. code-block:: php

$cfg['Servers'][$i]['ssl\_key'] = '/etc/mysql/server-key.pem';

.. config:option:: $cfg['Servers'][$i]['ssl\_cert']

:type: string

:default: NULL

Path to the cert file when using SSL for connecting to the MySQL server.

.. config:option:: $cfg['Servers'][$i]['ssl\_ca']

:type: string

:default: NULL

Path to the CA file when using SSL for connecting to the MySQL server.

.. config:option:: $cfg['Servers'][$i]['ssl\_ca\_path']

:type: string

:default: NULL

Directory containing trusted SSL CA certificates in PEM format.

.. config:option:: $cfg['Servers'][$i]['ssl\_ciphers']

:type: string

:default: NULL

List of allowable ciphers for SSL connections to the MySQL server.

.. config:option:: $cfg['Servers'][$i]['connect\_type']

:type: string

:default: ``'tcp'``

What type connection to use with the MySQL server. Your options are

``'socket'`` and ``'tcp'``. It defaults to tcp as that is nearly guaranteed

to be available on all MySQL servers, while sockets are not supported on

some platforms. To use the socket mode, your MySQL server must be on the

same machine as the Web server.

.. config:option:: $cfg['Servers'][$i]['compress']

:type: boolean

:default: false

Whether to use a compressed protocol for the MySQL server connection

or not (experimental).

.. \_controlhost:

.. config:option:: $cfg['Servers'][$i]['controlhost']

:type: string

:default: ``''``

Permits to use an alternate host to hold the configuration storage

data.

.. \_controlport:

.. config:option:: $cfg['Servers'][$i]['controlport']

:type: string

:default: ``''``

Permits to use an alternate port to connect to the host that

holds the configuration storage.

.. \_controluser:

.. config:option:: $cfg['Servers'][$i]['controluser']

:type: string

:default: ``''``

.. config:option:: $cfg['Servers'][$i]['controlpass']

:type: string

:default: ``''``

This special account is used for 2 distinct purposes: to make possible all

relational features (see :config:option:`$cfg['Servers'][$i]['pmadb']`).

.. versionchanged:: 2.2.5

those were called ``stduser`` and ``stdpass``

.. seealso:: :ref:`setup`, :ref:`authentication\_modes`, :ref:`linked-tables`

.. config:option:: $cfg['Servers'][$i]['auth\_type']

:type: string

:default: ``'cookie'``

Whether config or cookie or :term:`HTTP` or signon authentication should be

used for this server.

\* 'config' authentication (``$auth\_type = 'config'``) is the plain old

way: username and password are stored in :file:`config.inc.php`.

\* 'cookie' authentication mode (``$auth\_type = 'cookie'``) allows you to

log in as any valid MySQL user with the help of cookies.

\* 'http' authentication allows you to log in as any

valid MySQL user via HTTP-Auth.

\* 'signon' authentication mode (``$auth\_type = 'signon'``) allows you to

log in from prepared PHP session data or using supplied PHP script.

.. seealso:: :ref:`authentication\_modes`

.. \_servers\_auth\_http\_realm:

.. config:option:: $cfg['Servers'][$i]['auth\_http\_realm']

:type: string

:default: ``''``

When using auth\\_type = ``http``, this field allows to define a custom

:term:`HTTP` Basic Auth Realm which will be displayed to the user. If not

explicitly specified in your configuration, a string combined of

"phpMyAdmin " and either :config:option:`$cfg['Servers'][$i]['verbose']` or

:config:option:`$cfg['Servers'][$i]['host']` will be used.

.. \_servers\_auth\_swekey\_config:

.. config:option:: $cfg['Servers'][$i]['auth\_swekey\_config']

:type: string

:default: ``''``

The name of the file containing :ref:`swekey` ids and login names for hardware

authentication. Leave empty to deactivate this feature.

.. \_servers\_user:

.. config:option:: $cfg['Servers'][$i]['user']

:type: string

:default: ``'root'``

.. config:option:: $cfg['Servers'][$i]['password']

:type: string

:default: ``''``

When using :config:option:`$cfg['Servers'][$i]['auth\_type']` set to

'config', this is the user/password-pair which phpMyAdmin will use to

connect to the MySQL server. This user/password pair is not needed when

:term:`HTTP` or cookie authentication is used

and should be empty.

.. \_servers\_nopassword:

.. config:option:: $cfg['Servers'][$i]['nopassword']

:type: boolean

:default: false

Allow attempt to log in without password when a login with password

fails. This can be used together with http authentication, when

authentication is done some other way and phpMyAdmin gets user name

from auth and uses empty password for connecting to MySQL. Password

login is still tried first, but as fallback, no password method is

tried.

.. \_servers\_only\_db:

.. config:option:: $cfg['Servers'][$i]['only\_db']

:type: string or array

:default: ``''``

If set to a (an array of) database name(s), only this (these)

database(s) will be shown to the user. Since phpMyAdmin 2.2.1,

this/these database(s) name(s) may contain MySQL wildcards characters

("\\_" and "%"): if you want to use literal instances of these

characters, escape them (I.E. use ``'my\\_db'`` and not ``'my\_db'``).

This setting is an efficient way to lower the server load since the

latter does not need to send MySQL requests to build the available

database list. But \*\*it does not replace the privileges rules of the

MySQL database server\*\*. If set, it just means only these databases

will be displayed but \*\*not that all other databases can't be used.\*\*

An example of using more that one database:

.. code-block:: php

$cfg['Servers'][$i]['only\_db'] = array('db1', 'db2');

.. versionchanged:: 4.0.0

Previous versions permitted to specify the display order of

the database names via this directive.

.. config:option:: $cfg['Servers'][$i]['hide\_db']

:type: string

:default: ``''``

Regular expression for hiding some databases from unprivileged users.

This only hides them from listing, but a user is still able to access

them (using, for example, the SQL query area). To limit access, use

the MySQL privilege system. For example, to hide all databases

starting with the letter "a", use

.. code-block:: php

$cfg['Servers'][$i]['hide\_db'] = '^a';

and to hide both "db1" and "db2" use

.. code-block:: php

$cfg['Servers'][$i]['hide\_db'] = '^(db1|db2)$';

More information on regular expressions can be found in the `PCRE

pattern syntax

<http://php.net/manual/en/reference.pcre.pattern.syntax.php>`\_ portion

of the PHP reference manual.

.. config:option:: $cfg['Servers'][$i]['verbose']

:type: string

:default: ``''``

Only useful when using phpMyAdmin with multiple server entries. If

set, this string will be displayed instead of the hostname in the

pull-down menu on the main page. This can be useful if you want to

show only certain databases on your system, for example. For HTTP

auth, all non-US-ASCII characters will be stripped.

.. config:option:: $cfg['Servers'][$i]['pmadb']

:type: string

:default: ``''``

The name of the database containing the phpMyAdmin configuration

storage.

See the :ref:`linked-tables` section in this document to see the benefits of

this feature, and for a quick way of creating this database and the needed

tables.

If you are the only user of this phpMyAdmin installation, you can use your

current database to store those special tables; in this case, just put your

current database name in :config:option:`$cfg['Servers'][$i]['pmadb']`. For a

multi-user installation, set this parameter to the name of your central

database containing the phpMyAdmin configuration storage.

.. \_bookmark:

.. config:option:: $cfg['Servers'][$i]['bookmarktable']

:type: string

:default: ``''``

Since release 2.2.0 phpMyAdmin allows users to bookmark queries. This

can be useful for queries you often run. To allow the usage of this

functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* enter the table name in :config:option:`$cfg['Servers'][$i]['bookmarktable']`

.. \_relation:

.. config:option:: $cfg['Servers'][$i]['relation']

:type: string

:default: ``''``

Since release 2.2.4 you can describe, in a special 'relation' table,

which column is a key in another table (a foreign key). phpMyAdmin

currently uses this to:

\* make clickable, when you browse the master table, the data values that

point to the foreign table;

\* display in an optional tool-tip the "display column" when browsing the

master table, if you move the mouse to a column containing a foreign

key (use also the 'table\\_info' table); (see :ref:`faqdisplay`)

\* in edit/insert mode, display a drop-down list of possible foreign keys

(key value and "display column" are shown) (see :ref:`faq6\_21`)

\* display links on the table properties page, to check referential

integrity (display missing foreign keys) for each described key;

\* in query-by-example, create automatic joins (see :ref:`faq6\_6`)

\* enable you to get a :term:`PDF` schema of

your database (also uses the table\\_coords table).

The keys can be numeric or character.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the relation table name in :config:option:`$cfg['Servers'][$i]['relation']`

\* now as normal user open phpMyAdmin and for each one of your tables

where you want to use this feature, click :guilabel:`Structure/Relation view/`

and choose foreign columns.

.. note::

In the current version, ``master\_db`` must be the same as ``foreign\_db``.

Those columns have been put in future development of the cross-db

relations.

.. \_table\_info:

.. config:option:: $cfg['Servers'][$i]['table\_info']

:type: string

:default: ``''``

Since release 2.3.0 you can describe, in a special 'table\\_info'

table, which column is to be displayed as a tool-tip when moving the

cursor over the corresponding key. This configuration variable will

hold the name of this special table. To allow the usage of this

functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['table\\_info']` (e.g.

``pma\_\_table\_info``)

\* then for each table where you want to use this feature, click

"Structure/Relation view/Choose column to display" to choose the

column.

.. seealso:: :ref:`faqdisplay`

.. \_table\_coords:

.. config:option:: $cfg['Servers'][$i]['table\_coords']

:type: string

:default: ``''``

.. config:option:: $cfg['Servers'][$i]['pdf\_pages']

:type: string

:default: ``''``

Since release 2.3.0 you can have phpMyAdmin create :term:`PDF` pages

showing the relations between your tables. Further, the designer interface

permits visually managing the relations. To do this it needs two tables

"pdf\\_pages" (storing information about the available :term:`PDF` pages)

and "table\\_coords" (storing coordinates where each table will be placed on

a :term:`PDF` schema output). You must be using the "relation" feature.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the correct table names in

:config:option:`$cfg['Servers'][$i]['table\\_coords']` and

:config:option:`$cfg['Servers'][$i]['pdf\\_pages']`

.. seealso:: :ref:`faqpdf`.

.. \_col\_com:

.. config:option:: $cfg['Servers'][$i]['column\_info']

:type: string

:default: ``''``

This part requires a content update! Since release 2.3.0 you can

store comments to describe each column for each table. These will then

be shown on the "printview".

Starting with release 2.5.0, comments are consequently used on the table

property pages and table browse view, showing up as tool-tips above the

column name (properties page) or embedded within the header of table in

browse view. They can also be shown in a table dump. Please see the

relevant configuration directives later on.

Also new in release 2.5.0 is a MIME- transformation system which is also

based on the following table structure. See :ref:`transformations` for

further information. To use the MIME- transformation system, your

column\\_info table has to have the three new columns 'mimetype',

'transformation', 'transformation\\_options'.

Starting with release 4.3.0, a new input-oriented transformation system

has been introduced. Also, backward compatibility code used in the old

transformations system was removed. As a result, an update to column\\_info

table is necessary for previous transformations and the new input-oriented

transformation system to work. phpMyAdmin will upgrade it automatically

for you by analyzing your current column\\_info table structure.

However, if something goes wrong with the auto-upgrade then you can

use the SQL script found in ``./sql/upgrade\_column\_info\_4\_3\_0+.sql``

to upgrade it manually.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['column\\_info']` (e.g.

``pma\_\_column\_info``)

\* to update your PRE-2.5.0 Column\\_comments table use this: and

remember that the Variable in :file:`config.inc.php` has been renamed from

:config:option:`$cfg['Servers'][$i]['column\\_comments']` to

:config:option:`$cfg['Servers'][$i]['column\\_info']`

.. code-block:: mysql

ALTER TABLE `pma\_\_column\_comments`

ADD `mimetype` VARCHAR( 255 ) NOT NULL,

ADD `transformation` VARCHAR( 255 ) NOT NULL,

ADD `transformation\_options` VARCHAR( 255 ) NOT NULL;

\* to update your PRE-4.3.0 Column\\_info table manually use this

``./sql/upgrade\_column\_info\_4\_3\_0+.sql`` SQL script.

.. note::

For auto-upgrade functionality to work, your

``$cfg['Servers'][$i]['controluser']`` must have ALTER privilege on

``phpmyadmin`` database. See the `MySQL documentation for GRANT

<http://dev.mysql.com/doc/mysql/en/grant.html>`\_ on how to

``GRANT`` privileges to a user.

.. \_history:

.. config:option:: $cfg['Servers'][$i]['history']

:type: string

:default: ``''``

Since release 2.5.0 you can store your :term:`SQL` history, which means all

queries you entered manually into the phpMyAdmin interface. If you don't

want to use a table-based history, you can use the JavaScript-based

history.

Using that, all your history items are deleted when closing the window.

Using :config:option:`$cfg['QueryHistoryMax']` you can specify an amount of

history items you want to have on hold. On every login, this list gets cut

to the maximum amount.

The query history is only available if JavaScript is enabled in

your browser.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['history']` (e.g.

``pma\_\_history``)

.. \_recent:

.. config:option:: $cfg['Servers'][$i]['recent']

:type: string

:default: ``''``

Since release 3.5.0 you can show recently used tables in the

navigation panel. It helps you to jump across table directly, without

the need to select the database, and then select the table. Using

:config:option:`$cfg['NumRecentTables']` you can configure the maximum number

of recent tables shown. When you select a table from the list, it will jump to

the page specified in :config:option:`$cfg['NavigationTreeDefaultTabTable']`.

Without configuring the storage, you can still access the recently used tables,

but it will disappear after you logout.

To allow the usage of this functionality persistently:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['recent']` (e.g.

``pma\_\_recent``)

.. \_table\_uiprefs:

.. config:option:: $cfg['Servers'][$i]['table\_uiprefs']

:type: string

:default: ``''``

Since release 3.5.0 phpMyAdmin can be configured to remember several

things (sorted column :config:option:`$cfg['RememberSorting']`, column order,

and column visibility from a database table) for browsing tables. Without

configuring the storage, these features still can be used, but the values will

disappear after you logout.

To allow the usage of these functionality persistently:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['table\\_uiprefs']` (e.g.

``pma\_\_table\_uiprefs``)

.. \_configurablemenus:

.. config:option:: $cfg['Servers'][$i]['users']

:type: string

:default: ``''``

.. config:option:: $cfg['Servers'][$i]['usergroups']

:type: string

:default: ``''``

Since release 4.1.0 you can create different user groups with menu items

attached to them. Users can be assigned to these groups and the logged in

user would only see menu items configured to the usergroup he is assigned to.

To do this it needs two tables "usergroups" (storing allowed menu items for each

user group) and "users" (storing users and their assignments to user groups).

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the correct table names in

:config:option:`$cfg['Servers'][$i]['users']` (e.g. ``pma\_\_users``) and

:config:option:`$cfg['Servers'][$i]['usergroups']` (e.g. ``pma\_\_usergroups``)

.. \_navigationhiding:

.. config:option:: $cfg['Servers'][$i]['navigationhiding']

:type: string

:default: ``''``

Since release 4.1.0 you can hide/show items in the navigation tree.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['navigationhiding']` (e.g.

``pma\_\_navigationhiding``)

.. \_central\_columns:

.. config:option:: $cfg['Servers'][$i]['central\_columns']

:type: string

:default: ``''``

Since release 4.3.0 you can have a central list of columns per database.

You can add/remove columns to the list as per your requirement. These columns

in the central list will be available to use while you create a new column for

a table or create a table itself. You can select a column from central list

while creating a new column, it will save you from writing the same column definition

over again or from writing different names for similar column.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['central\_columns']` (e.g.

``pma\_\_central\_columns``)

.. \_designer\_settings:

.. config:option:: $cfg['Servers'][$i]['designer\_settings']

:type: string

:default: ``''``

Since release 4.5.0 your designer settings can be remembered.

Your choice regarding 'Angular/Direct Links', 'Snap to Grid', 'Toggle Relation Lines',

'Small/Big All', 'Move Menu' and 'Pin Text' can be remembered persistently.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['designer\_settings']` (e.g.

``pma\_\_designer\_settings``)

.. \_savedsearches:

.. config:option:: $cfg['Servers'][$i]['savedsearches']

:type: string

:default: ``''``

Since release 4.2.0 you can save and load query-by-example searches from the Database > Query panel.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['savedsearches']` (e.g.

``pma\_\_savedsearches``)

.. \_export\_templates:

.. config:option:: $cfg['Servers'][$i]['export\_templates']

:type: string

:default: ``''``

Since release 4.5.0 you can save and load export templates.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['export\_templates']` (e.g.

``pma\_\_export\_templates``)

.. \_tracking:

.. config:option:: $cfg['Servers'][$i]['tracking']

:type: string

:default: ``''``

Since release 3.3.x a tracking mechanism is available. It helps you to

track every :term:`SQL` command which is

executed by phpMyAdmin. The mechanism supports logging of data

manipulation and data definition statements. After enabling it you can

create versions of tables.

The creation of a version has two effects:

\* phpMyAdmin saves a snapshot of the table, including structure and

indexes.

\* phpMyAdmin logs all commands which change the structure and/or data of

the table and links these commands with the version number.

Of course you can view the tracked changes. On the :guilabel:`Tracking`

page a complete report is available for every version. For the report you

can use filters, for example you can get a list of statements within a date

range. When you want to filter usernames you can enter \\* for all names or

you enter a list of names separated by ','. In addition you can export the

(filtered) report to a file or to a temporary database.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['tracking']` (e.g.

``pma\_\_tracking``)

.. \_tracking2:

.. config:option:: $cfg['Servers'][$i]['tracking\_version\_auto\_create']

:type: boolean

:default: false

Whether the tracking mechanism creates versions for tables and views

automatically.

If this is set to true and you create a table or view with

\* CREATE TABLE ...

\* CREATE VIEW ...

and no version exists for it, the mechanism will create a version for

you automatically.

.. \_tracking3:

.. config:option:: $cfg['Servers'][$i]['tracking\_default\_statements']

:type: string

:default: ``'CREATE TABLE,ALTER TABLE,DROP TABLE,RENAME TABLE,CREATE INDEX,DROP INDEX,INSERT,UPDATE,DELETE,TRUNCATE,REPLACE,CREATE VIEW,ALTER VIEW,DROP VIEW,CREATE DATABASE,ALTER DATABASE,DROP DATABASE'``

Defines the list of statements the auto-creation uses for new

versions.

.. \_tracking4:

.. config:option:: $cfg['Servers'][$i]['tracking\_add\_drop\_view']

:type: boolean

:default: true

Whether a DROP VIEW IF EXISTS statement will be added as first line to

the log when creating a view.

.. \_tracking5:

.. config:option:: $cfg['Servers'][$i]['tracking\_add\_drop\_table']

:type: boolean

:default: true

Whether a DROP TABLE IF EXISTS statement will be added as first line

to the log when creating a table.

.. \_tracking6:

.. config:option:: $cfg['Servers'][$i]['tracking\_add\_drop\_database']

:type: boolean

:default: true

Whether a DROP DATABASE IF EXISTS statement will be added as first

line to the log when creating a database.

.. \_userconfig:

.. config:option:: $cfg['Servers'][$i]['userconfig']

:type: string

:default: ``''``

Since release 3.4.x phpMyAdmin allows users to set most preferences by

themselves and store them in the database.

If you don't allow for storing preferences in

:config:option:`$cfg['Servers'][$i]['pmadb']`, users can still personalize

phpMyAdmin, but settings will be saved in browser's local storage, or, it

is is unavailable, until the end of session.

To allow the usage of this functionality:

\* set up :config:option:`$cfg['Servers'][$i]['pmadb']` and the phpMyAdmin configuration storage

\* put the table name in :config:option:`$cfg['Servers'][$i]['userconfig']`

.. config:option:: $cfg['Servers'][$i]['MaxTableUiprefs']

:type: integer

:default: 100

Maximum number of rows saved in

:config:option:`$cfg['Servers'][$i]['table\_uiprefs']` table.

When tables are dropped or renamed,

:config:option:`$cfg['Servers'][$i]['table\_uiprefs']` may contain invalid data

(referring to tables which no longer exist). We only keep this number of newest

rows in :config:option:`$cfg['Servers'][$i]['table\_uiprefs']` and automatically

delete older rows.

.. config:option:: $cfg['Servers'][$i]['SessionTimeZone']

:type: string

:default: ``''``

Sets the time zone used by phpMyAdmin. Leave blank to use the time zone of your

database server. Possible values are explained at

http://dev.mysql.com/doc/refman/5.7/en/time-zone-support.html

This is useful when your database server uses a time zone which is different from the

time zone you want to use in phpMyAdmin.

.. config:option:: $cfg['Servers'][$i]['AllowRoot']

:type: boolean

:default: true

Whether to allow root access. This is just a shortcut for the

:config:option:`$cfg['Servers'][$i]['AllowDeny']['rules']` below.

.. config:option:: $cfg['Servers'][$i]['AllowNoPassword']

:type: boolean

:default: false

Whether to allow logins without a password. The default value of

``false`` for this parameter prevents unintended access to a MySQL

server with was left with an empty password for root or on which an

anonymous (blank) user is defined.

.. \_servers\_allowdeny\_order:

.. config:option:: $cfg['Servers'][$i]['AllowDeny']['order']

:type: string

:default: ``''``

If your rule order is empty, then :term:`IP`

authorization is disabled.

If your rule order is set to

``'deny,allow'`` then the system applies all deny rules followed by

allow rules. Access is allowed by default. Any client which does not

match a Deny command or does match an Allow command will be allowed

access to the server.

If your rule order is set to ``'allow,deny'``

then the system applies all allow rules followed by deny rules. Access

is denied by default. Any client which does not match an Allow

directive or does match a Deny directive will be denied access to the

server.

If your rule order is set to ``'explicit'``, authorization is

performed in a similar fashion to rule order 'deny,allow', with the

added restriction that your host/username combination \*\*must\*\* be

listed in the \*allow\* rules, and not listed in the \*deny\* rules. This

is the \*\*most\*\* secure means of using Allow/Deny rules, and was

available in Apache by specifying allow and deny rules without setting

any order.

Please also see :config:option:`$cfg['TrustedProxies']` for

detecting IP address behind proxies.

.. \_servers\_allowdeny\_rules:

.. config:option:: $cfg['Servers'][$i]['AllowDeny']['rules']

:type: array of strings

:default: array()

The general format for the rules is as such:

.. code-block:: none

<'allow' | 'deny'> <username> [from] <ipmask>

If you wish to match all users, it is possible to use a ``'%'`` as a

wildcard in the \*username\* field.

There are a few shortcuts you can

use in the \*ipmask\* field as well (please note that those containing

SERVER\\_ADDRESS might not be available on all webservers):

.. code-block:: none

'all' -> 0.0.0.0/0

'localhost' -> 127.0.0.1/8

'localnetA' -> SERVER\_ADDRESS/8

'localnetB' -> SERVER\_ADDRESS/16

'localnetC' -> SERVER\_ADDRESS/24

Having an empty rule list is equivalent to either using ``'allow %

from all'`` if your rule order is set to ``'deny,allow'`` or ``'deny %

from all'`` if your rule order is set to ``'allow,deny'`` or

``'explicit'``.

For the :term:`IP address` matching

system, the following work:

\* ``xxx.xxx.xxx.xxx`` (an exact :term:`IP address`)

\* ``xxx.xxx.xxx.[yyy-zzz]`` (an :term:`IP address` range)

\* ``xxx.xxx.xxx.xxx/nn`` (CIDR, Classless Inter-Domain Routing type :term:`IP` addresses)

But the following does not work:

\* ``xxx.xxx.xxx.xx[yyy-zzz]`` (partial :term:`IP` address range)

For :term:`IPv6` addresses, the following work:

\* ``xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx`` (an exact :term:`IPv6` address)

\* ``xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:[yyyy-zzzz]`` (an :term:`IPv6` address range)

\* ``xxxx:xxxx:xxxx:xxxx/nn`` (CIDR, Classless Inter-Domain Routing type :term:`IPv6` addresses)

But the following does not work:

\* ``xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xx[yyy-zzz]`` (partial :term:`IPv6` address range)

.. config:option:: $cfg['Servers'][$i]['DisableIS']

:type: boolean

:default: false

Disable using ``INFORMATION\_SCHEMA`` to retrieve information (use

``SHOW`` commands instead), because of speed issues when many

databases are present. Currently used in some parts of the code, more

to come.

.. config:option:: $cfg['Servers'][$i]['SignonScript']

:type: string

:default: ``''``

Name of PHP script to be sourced and executed to obtain login

credentials. This is alternative approach to session based single

signon. The script has to provide a function called

``get\_login\_credentials`` which returns list of username and

password, accepting single parameter of existing username (can be

empty). See :file:`examples/signon-script.php` for an example:

.. literalinclude:: ../examples/signon-script.php

:language: php

.. seealso:: :ref:`auth\_signon`

.. config:option:: $cfg['Servers'][$i]['SignonSession']

:type: string

:default: ``''``

Name of session which will be used for signon authentication method.

You should use something different than ``phpMyAdmin``, because this

is session which phpMyAdmin uses internally. Takes effect only if

:config:option:`$cfg['Servers'][$i]['SignonScript']` is not configured.

.. seealso:: :ref:`auth\_signon`

.. config:option:: $cfg['Servers'][$i]['SignonURL']

:type: string

:default: ``''``

:term:`URL` where user will be redirected

to log in for signon authentication method. Should be absolute

including protocol.

.. seealso:: :ref:`auth\_signon`

.. config:option:: $cfg['Servers'][$i]['LogoutURL']

:type: string

:default: ``''``

:term:`URL` where user will be redirected

after logout (doesn't affect config authentication method). Should be

absolute including protocol.

Generic settings

----------------

.. config:option:: $cfg['ServerDefault']

:type: integer

:default: 1

If you have more than one server configured, you can set

:config:option:`$cfg['ServerDefault']` to any one of them to autoconnect to that

server when phpMyAdmin is started, or set it to 0 to be given a list

of servers without logging in.

If you have only one server configured,

:config:option:`$cfg['ServerDefault']` MUST be set to that server.

.. config:option:: $cfg['VersionCheck']

:type: boolean

:default: true

Enables check for latest versions using JavaScript on the main phpMyAdmin

page or by directly accessing :file:`version\_check.php`.

.. note::

This setting can be adjusted by your vendor.

.. config:option:: $cfg['ProxyUrl']

:type: string

:default: ""

The url of the proxy to be used when phpmyadmin needs to access the outside

internet such as when retrieving the latest version info or submitting error

reports. You need this if the server where phpMyAdmin is installed does not

have direct access to the internet.

The format is: "hostname:portnumber"

.. config:option:: $cfg['ProxyUser']

:type: string

:default: ""

The username for authenticating with the proxy. By default, no

authentication is performed. If a username is supplied, Basic

Authentication will be performed. No other types of authentication

are currently supported.

.. config:option:: $cfg['ProxyPass']

:type: string

:default: ""

The password for authenticating with the proxy.

.. config:option:: $cfg['MaxDbList']

:type: integer

:default: 100

The maximum number of database names to be displayed in the main panel's

database list.

.. config:option:: $cfg['MaxTableList']

:type: integer

:default: 250

The maximum number of table names to be displayed in the main panel's

list (except on the Export page).

.. config:option:: $cfg['ShowHint']

:type: boolean

:default: true

Whether or not to show hints (for example, hints when hovering over

table headers).

.. config:option:: $cfg['MaxCharactersInDisplayedSQL']

:type: integer

:default: 1000

The maximum number of characters when a :term:`SQL` query is displayed. The

default limit of 1000 should be correct to avoid the display of tons of

hexadecimal codes that represent BLOBs, but some users have real

:term:`SQL` queries that are longer than 1000 characters. Also, if a

query's length exceeds this limit, this query is not saved in the history.

.. config:option:: $cfg['PersistentConnections']

:type: boolean

:default: false

Whether `persistent connections <http://php.net/manual/en/features

.persistent-connections.php>`\_ should be used or not. Works with

following extensions:

\* mysql (`mysql\\_pconnect <http://php.net/manual/en/function.mysql-

pconnect.php>`\_),

\* mysqli (requires PHP 5.3.0 or newer, `more information

<http://php.net/manual/en/mysqli.persistconns.php>`\_).

.. config:option:: $cfg['ForceSSL']

:type: boolean

:default: false

Whether to force using https while accessing phpMyAdmin. In a reverse

proxy setup, setting this to ``true`` is not supported.

.. note::

In some setups (like separate SSL proxy or load balancer) you might

have to set :config:option:`$cfg['PmaAbsoluteUri']` for correct

redirection.

.. config:option:: $cfg['ExecTimeLimit']

:type: integer [number of seconds]

:default: 300

Set the number of seconds a script is allowed to run. If seconds is

set to zero, no time limit is imposed. This setting is used while

importing/exporting dump files but has

no effect when PHP is running in safe mode.

.. config:option:: $cfg['SessionSavePath']

:type: string

:default: ``''``

Path for storing session data (`session\\_save\\_path PHP parameter

<http://php.net/session\_save\_path>`\_).

.. config:option:: $cfg['MemoryLimit']

:type: string [number of bytes]

:default: ``'-1'``

Set the number of bytes a script is allowed to allocate. If set to

``'-1'``, no limit is imposed. If set to ``'0'``, no change of the

memory limit is attempted and the :file:`php.ini` ``memory\_limit`` is

used.

This setting is used while importing/exporting dump files

so you definitely don't want to put here a too low

value. It has no effect when PHP is running in safe mode.

You can also use any string as in :file:`php.ini`, eg. '16M'. Ensure you

don't omit the suffix (16 means 16 bytes!)

.. config:option:: $cfg['SkipLockedTables']

:type: boolean

:default: false

Mark used tables and make it possible to show databases with locked

tables (since MySQL 3.23.30).

.. config:option:: $cfg['ShowSQL']

:type: boolean

:default: true

Defines whether :term:`SQL` queries

generated by phpMyAdmin should be displayed or not.

.. config:option:: $cfg['RetainQueryBox']

:type: boolean

:default: false

Defines whether the :term:`SQL` query box

should be kept displayed after its submission.

.. config:option:: $cfg['CodemirrorEnable']

:type: boolean

:default: true

Defines whether to use a Javascript code editor for SQL query boxes.

CodeMirror provides syntax highlighting and line numbers. However,

middle-clicking for pasting the clipboard contents in some Linux

distributions (such as Ubuntu) is not supported by all browsers.

.. config:option:: $cfg['DefaultForeignKeyChecks']

:type: string

:default: ``'default'``

Default value of the checkbox for foreign key checks, to disable/enable

foreign key checks for certain queries. The possible values are ``'default'``,

``'enable'`` or ``'disable'``. If set to ``'default'``, the value of the

MySQL variable ``FOREIGN\_KEY\_CHECKS`` is used.

.. config:option:: $cfg['AllowUserDropDatabase']

:type: boolean

:default: false

Defines whether normal users (non-administrator) are allowed to delete

their own database or not. If set as false, the link :guilabel:`Drop

Database` will not be shown, and even a ``DROP DATABASE mydatabase`` will

be rejected. Quite practical for :term:`ISP` 's with many customers.

.. note::

This limitation of :term:`SQL` queries is not

as strict as when using MySQL privileges. This is due to nature of

:term:`SQL` queries which might be quite

complicated. So this choice should be viewed as help to avoid accidental

dropping rather than strict privilege limitation.

.. config:option:: $cfg['Confirm']

:type: boolean

:default: true

Whether a warning ("Are your really sure...") should be displayed when

you're about to lose data.

.. config:option:: $cfg['UseDbSearch']

:type: boolean

:default: true

Define whether the "search string inside database" is enabled or not.

.. config:option:: $cfg['IgnoreMultiSubmitErrors']

:type: boolean

:default: false

Define whether phpMyAdmin will continue executing a multi-query

statement if one of the queries fails. Default is to abort execution.

Cookie authentication options

-----------------------------

.. config:option:: $cfg['blowfish\_secret']

:type: string

:default: ``''``

The "cookie" auth\\_type uses AES algorithm to encrypt the password. If you

are using the "cookie" auth\\_type, enter here a random passphrase of your

choice. It will be used internally by the AES algorithm: you won’t be

prompted for this passphrase. There is no maximum length for this secret.

.. note::

The configuration is called blowfish\_secret for historical reasons as

Blowfish algorithm was originally used to do the encryption.

.. versionchanged:: 3.1.0

Since version 3.1.0 phpMyAdmin can generate this on the fly, but it

makes a bit weaker security as this generated secret is stored in

session and furthermore it makes impossible to recall user name from

cookie.

.. config:option:: $cfg['LoginCookieRecall']

:type: boolean

:default: true

Define whether the previous login should be recalled or not in cookie

authentication mode.

This is automatically disabled if you do not have

configured :config:option:`$cfg['blowfish\_secret']`.

.. config:option:: $cfg['LoginCookieValidity']

:type: integer [number of seconds]

:default: 1440

Define how long a login cookie is valid. Please note that php

configuration option `session.gc\\_maxlifetime

<http://php.net/manual/en/session.configuration.php#ini.session.gc-

maxlifetime>`\_ might limit session validity and if the session is lost,

the login cookie is also invalidated. So it is a good idea to set

``session.gc\_maxlifetime`` at least to the same value of

:config:option:`$cfg['LoginCookieValidity']`.

.. config:option:: $cfg['LoginCookieStore']

:type: integer [number of seconds]

:default: 0

Define how long login cookie should be stored in browser. Default 0

means that it will be kept for existing session. This is recommended

for not trusted environments.

.. config:option:: $cfg['LoginCookieDeleteAll']

:type: boolean

:default: true

If enabled (default), logout deletes cookies for all servers,

otherwise only for current one. Setting this to false makes it easy to

forget to log out from other server, when you are using more of them.

.. \_AllowArbitraryServer:

.. config:option:: $cfg['AllowArbitraryServer']

:type: boolean

:default: false

If enabled, allows you to log in to arbitrary servers using cookie

authentication.

.. note::

Please use this carefully, as this may allow users access to MySQL servers

behind the firewall where your :term:`HTTP` server is placed.

See also :config:option:`$cfg['ArbitraryServerRegexp']`.

.. config:option:: $cfg['ArbitraryServerRegexp']

:type: string

:default: ``''``

Restricts the MySQL servers to which the user can log in when

:config:option:`$cfg['AllowArbitraryServer']` is enabled by

matching the :term:`IP` or the hostname of the MySQL server

to the given regular expression. The regular expression must be enclosed

with a delimiter character.

.. config:option:: $cfg['CaptchaLoginPublicKey']

:type: string

:default: ``''``

The public key for the reCaptcha service that can be obtained from

http://www.google.com/recaptcha.

reCaptcha will be then used in :ref:`cookie`.

.. config:option:: $cfg['CaptchaLoginPrivateKey']

:type: string

:default: ``''``

The private key for the reCaptcha service that can be obtain from

http://www.google.com/recaptcha.

reCaptcha will be then used in :ref:`cookie`.

Navigation panel setup

----------------------

.. config:option:: $cfg['ShowDatabasesNavigationAsTree']

:type: boolean

:default: true

In the navigation panel, replaces the database tree with a selector

.. config:option:: $cfg['FirstLevelNavigationItems']

:type: integer

:default: 100

The number of first level databases that can be displayed on each page

of navigation tree.

.. config:option:: $cfg['MaxNavigationItems']

:type: integer

:default: 50

The number of items (tables, columns, indexes) that can be displayed on each

page of the navigation tree.

.. config:option:: $cfg['NavigationTreeEnableGrouping']

:type: boolean

:default: true

Defines whether to group the databases based on a common prefix

in their name :config:option:`$cfg['NavigationTreeDbSeparator']`.

.. config:option:: $cfg['NavigationTreeDbSeparator']

:type: string or array

:default: ``'\_'``

The string used to separate the parts of the database name when

showing them in a tree. Alternatively you can specify more strings in

an array and all of them will be used as a separator.

.. config:option:: $cfg['NavigationTreeTableSeparator']

:type: string or array

:default: ``'\_\_'``

Defines a string to be used to nest table spaces. This means if you have

tables like ``first\_\_second\_\_third`` this will be shown as a three-level

hierarchy like: first > second > third. If set to false or empty, the

feature is disabled. NOTE: You should not use this separator at the

beginning or end of a table name or multiple times after another without

any other characters in between.

.. config:option:: $cfg['NavigationTreeTableLevel']

:type: integer

:default: 1

Defines how many sublevels should be displayed when splitting up

tables by the above separator.

.. config:option:: $cfg['NumRecentTables']

:type: integer

:default: 10

The maximum number of recently used tables shown in the navigation

panel. Set this to 0 (zero) to disable the listing of recent tables.

.. config:option:: $cfg['ZeroConf']

:type: boolean

:default: true

Enables Zero Configuration mode in which the user will be offered a choice to

create phpMyAdmin configuration storage in the current database

or use the existing one, if already present.

This setting has no effect if the phpMyAdmin configuration storage database

is properly created and the related configuration directives (such as

:config:option:`$cfg['Servers'][$i]['pmadb']` and so on) are configured.

.. config:option:: $cfg['NavigationLinkWithMainPanel']

:type: boolean

:default: true

Defines whether or not to link with main panel by highlighting

the current database or table.

.. config:option:: $cfg['NavigationDisplayLogo']

:type: boolean

:default: true

Defines whether or not to display the phpMyAdmin logo at the top of

the navigation panel.

.. config:option:: $cfg['NavigationLogoLink']

:type: string

:default: ``'index.php'``

Enter :term:`URL` where logo in the navigation panel will point to.

For use especially with self made theme which changes this.

For external URLs, you should include URL scheme as well.

.. config:option:: $cfg['NavigationLogoLinkWindow']

:type: string

:default: ``'main'``

Whether to open the linked page in the main window (``main``) or in a

new one (``new``). Note: use ``new`` if you are linking to

``phpmyadmin.net``.

.. config:option:: $cfg['NavigationTreeDisplayItemFilterMinimum']

:type: integer

:default: 30

Defines the minimum number of items (tables, views, routines and

events) to display a JavaScript filter box above the list of items in

the navigation tree.

To disable the filter completely some high number can be used (e.g. 9999)

.. config:option:: $cfg['NavigationTreeDisplayDbFilterMinimum']

:type: integer

:default: 30

Defines the minimum number of databases to display a JavaScript filter

box above the list of databases in the navigation tree.

To disable the filter completely some high number can be used

(e.g. 9999)

.. config:option:: $cfg['NavigationDisplayServers']

:type: boolean

:default: true

Defines whether or not to display a server choice at the top of the

navigation panel.

.. config:option:: $cfg['DisplayServersList']

:type: boolean

:default: false

Defines whether to display this server choice as links instead of in a

drop-down.

.. config:option:: $cfg['NavigationTreeDefaultTabTable']

:type: string

:default: ``'structure'``

Defines the tab displayed by default when clicking the small icon next

to each table name in the navigation panel. The possible values are the

localized equivalent of:

\* ``structure``

\* ``sql``

\* ``search``

\* ``insert``

\* ``browse``

.. config:option:: $cfg['NavigationTreeDefaultTabTable2']

:type: string

:default: null

Defines the tab displayed by default when clicking the second small icon next

to each table name in the navigation panel. The possible values are the

localized equivalent of:

\* ``(empty)``

\* ``structure``

\* ``sql``

\* ``search``

\* ``insert``

\* ``browse``

.. config:option:: $cfg['NavigationTreeEnableExpansion']

:type: boolean

:default: false

Whether to offer the possibility of tree expansion in the navigation panel.

.. config:option:: $cfg['NavigationTreeShowTables']

:type: boolean

:default: true

Whether to show tables under database in the navigation panel.

.. config:option:: $cfg['NavigationTreeShowViews']

:type: boolean

:default: true

Whether to show views under database in the navigation panel.

.. config:option:: $cfg['NavigationTreeShowFunctions']

:type: boolean

:default: true

Whether to show functions under database in the navigation panel.

.. config:option:: $cfg['NavigationTreeShowProcedures']

:type: boolean

:default: true

Whether to show procedures under database in the navigation panel.

.. config:option:: $cfg['NavigationTreeShowEvents']

:type: boolean

:default: true

Whether to show events under database in the navigation panel.

Main panel

----------

.. config:option:: $cfg['ShowStats']

:type: boolean

:default: true

Defines whether or not to display space usage and statistics about

databases and tables. Note that statistics requires at least MySQL

3.23.3 and that, at this date, MySQL doesn't return such information

for Berkeley DB tables.

.. config:option:: $cfg['ShowServerInfo']

:type: boolean

:default: true

Defines whether to display detailed server information on main page.

You can additionally hide more information by using

:config:option:`$cfg['Servers'][$i]['verbose']`.

.. config:option:: $cfg['ShowPhpInfo']

:type: boolean

:default: false

.. config:option:: $cfg['ShowChgPassword']

:type: boolean

:default: true

.. config:option:: $cfg['ShowCreateDb']

:type: boolean

:default: true

Defines whether to display the :guilabel:`PHP information` and

:guilabel:`Change password` links and form for creating database or not at

the starting main (right) frame. This setting does not check MySQL commands

entered directly.

Please note that to block the usage of ``phpinfo()`` in scripts, you have to

put this in your :file:`php.ini`:

.. code-block:: ini

disable\_functions = phpinfo()

Also note that enabling the :guilabel:`Change password` link has no effect

with config authentication mode: because of the hard coded password value

in the configuration file, end users can't be allowed to change their

passwords.

.. config:option:: $cfg['ShowGitRevision']

:type: boolean

:default: true

Defines whether to display informations about the current Git revision (if

applicable) on the main panel.

.. config:option:: $cfg['MysqlMinVersion']

:type: array

Defines the minimum supported MySQL version. The default is chosen

by the phpMyAdmin team; however this directive was asked by a developer

of the Plesk control panel to ease integration with older MySQL servers

(where most of the phpMyAdmin features work).

Database structure

------------------

.. config:option:: $cfg['ShowDbStructureCreation']

:type: boolean

:default: false

Defines whether the database structure page (tables list) has a

"Creation" column that displays when each table was created.

.. config:option:: $cfg['ShowDbStructureLastUpdate']

:type: boolean

:default: false

Defines whether the database structure page (tables list) has a "Last

update" column that displays when each table was last updated.

.. config:option:: $cfg['ShowDbStructureLastCheck']

:type: boolean

:default: false

Defines whether the database structure page (tables list) has a "Last

check" column that displays when each table was last checked.

.. config:option:: $cfg['HideStructureActions']

:type: boolean

:default: true

Defines whether the table structure actions are hidden under a "More"

drop-down.

Browse mode

-----------

.. config:option:: $cfg['TableNavigationLinksMode']

:type: string

:default: ``'icons'``

Defines whether the table navigation links contain ``'icons'``, ``'text'``

or ``'both'``.

.. config:option:: $cfg['ActionLinksMode']

:type: string

:default: ``'both'``

If set to ``icons``, will display icons instead of text for db and table

properties links (like :guilabel:`Browse`, :guilabel:`Select`,

:guilabel:`Insert`, ...). Can be set to ``'both'``

if you want icons AND text. When set to ``text``, will only show text.

.. config:option:: $cfg['RowActionType']

:type: string

:default: ``'both'``

Whether to display icons or text or both icons and text in table row action

segment. Value can be either of ``'icons'``, ``'text'`` or ``'both'``.

.. config:option:: $cfg['ShowAll']

:type: boolean

:default: false

Defines whether a user should be displayed a "Show all" button in browse

mode or not in all cases. By default it is shown only on small tables (less

than 500 rows) to avoid performance issues while getting too many rows.

.. config:option:: $cfg['MaxRows']

:type: integer

:default: 25

Number of rows displayed when browsing a result set and no LIMIT

clause is used. If the result set contains more rows, "Previous" and

"Next" links will be shown. Possible values: 25,50,100,250,500.

.. config:option:: $cfg['Order']

:type: string

:default: ``'SMART'``

Defines whether columns are displayed in ascending (``ASC``) order, in

descending (``DESC``) order or in a "smart" (``SMART``) order - I.E.

descending order for columns of type TIME, DATE, DATETIME and

TIMESTAMP, ascending order else- by default.

.. config:option:: $cfg['GridEditing']

:type: string

:default: ``'double-click'``

Defines which action (``double-click`` or ``click``) triggers grid

editing. Can be deactivated with the ``disabled`` value.

.. config:option:: $cfg['RelationalDisplay']

:type: string

:default: ``'K'``

Defines the initial behavior for Options > Relational. ``K``, which

is the default, displays the key while ``D`` shows the display column.

.. config:option:: $cfg['SaveCellsAtOnce']

:type: boolean

:default: false

Defines whether or not to save all edited cells at once for grid

editing.

Editing mode

------------

.. config:option:: $cfg['ProtectBinary']

:type: boolean or string

:default: ``'blob'``

Defines whether ``BLOB`` or ``BINARY`` columns are protected from

editing when browsing a table's content. Valid values are:

\* ``false`` to allow editing of all columns;

\* ``'blob'`` to allow editing of all columns except ``BLOBS``;

\* ``'noblob'`` to disallow editing of all columns except ``BLOBS`` (the

opposite of ``'blob'``);

\* ``'all'`` to disallow editing of all ``BINARY`` or ``BLOB`` columns.

.. config:option:: $cfg['ShowFunctionFields']

:type: boolean

:default: true

Defines whether or not MySQL functions fields should be initially

displayed in edit/insert mode. Since version 2.10, the user can toggle

this setting from the interface.

.. config:option:: $cfg['ShowFieldTypesInDataEditView']

:type: boolean

:default: true

Defines whether or not type fields should be initially displayed in

edit/insert mode. The user can toggle this setting from the interface.

.. config:option:: $cfg['InsertRows']

:type: integer

:default: 2

Defines the maximum number of concurrent entries for the Insert page.

.. config:option:: $cfg['ForeignKeyMaxLimit']

:type: integer

:default: 100

If there are fewer items than this in the set of foreign keys, then a

drop-down box of foreign keys is presented, in the style described by

the :config:option:`$cfg['ForeignKeyDropdownOrder']` setting.

.. config:option:: $cfg['ForeignKeyDropdownOrder']

:type: array

:default: array('content-id', 'id-content')

For the foreign key drop-down fields, there are several methods of

display, offering both the key and value data. The contents of the

array should be one or both of the following strings: ``content-id``,

``id-content``.

Export and import settings

--------------------------

.. config:option:: $cfg['ZipDump']

:type: boolean

:default: true

.. config:option:: $cfg['GZipDump']

:type: boolean

:default: true

.. config:option:: $cfg['BZipDump']

:type: boolean

:default: true

Defines whether to allow the use of zip/GZip/BZip2 compression when

creating a dump file

.. config:option:: $cfg['CompressOnFly']

:type: boolean

:default: true

Defines whether to allow on the fly compression for GZip/BZip2

compressed exports. This doesn't affect smaller dumps and allows users

to create larger dumps that won't otherwise fit in memory due to php

memory limit. Produced files contain more GZip/BZip2 headers, but all

normal programs handle this correctly.

.. config:option:: $cfg['Export']

:type: array

:default: array(...)

In this array are defined default parameters for export, names of

items are similar to texts seen on export page, so you can easily

identify what they mean.

.. config:option:: $cfg['Export']['method']

:type: string

:default: ``'quick'``

Defines how the export form is displayed when it loads. Valid values

are:

\* ``quick`` to display the minimum number of options to configure

\* ``custom`` to display every available option to configure

\* ``custom-no-form`` same as ``custom`` but does not display the option

of using quick export

.. config:option:: $cfg['Import']

:type: array

:default: array(...)

In this array are defined default parameters for import, names of

items are similar to texts seen on import page, so you can easily

identify what they mean.

Tabs display settings

---------------------

.. config:option:: $cfg['TabsMode']

:type: string

:default: ``'both'``

Defines whether the menu tabs contain ``'icons'``, ``'text'`` or ``'both'``.

.. config:option:: $cfg['PropertiesNumColumns']

:type: integer

:default: 1

How many columns will be utilized to display the tables on the database

property view? When setting this to a value larger than 1, the type of the

database will be omitted for more display space.

.. config:option:: $cfg['DefaultTabServer']

:type: string

:default: ``'welcome'``

Defines the tab displayed by default on server view. The possible values

are the localized equivalent of:

\* ``welcome`` (recommended for multi-user setups)

\* ``databases``,

\* ``status``

\* ``variables``

\* ``privileges``

.. config:option:: $cfg['DefaultTabDatabase']

:type: string

:default: ``'structure'``

Defines the tab displayed by default on database view. The possible values

are the localized equivalent of:

\* ``structure``

\* ``sql``

\* ``search``

\* ``operations``

.. config:option:: $cfg['DefaultTabTable']

:type: string

:default: ``'browse'``

Defines the tab displayed by default on table view. The possible values

are the localized equivalent of:

\* ``structure``

\* ``sql``

\* ``search``

\* ``insert``

\* ``browse``

PDF Options

-----------

.. config:option:: $cfg['PDFPageSizes']

:type: array

:default: ``array('A3', 'A4', 'A5', 'letter', 'legal')``

Array of possible paper sizes for creating PDF pages.

You should never need to change this.

.. config:option:: $cfg['PDFDefaultPageSize']

:type: string

:default: ``'A4'``

Default page size to use when creating PDF pages. Valid values are any

listed in :config:option:`$cfg['PDFPageSizes']`.

Languages

---------

.. config:option:: $cfg['DefaultLang']

:type: string

:default: ``'en'``

Defines the default language to use, if not browser-defined or user-

defined. The corresponding language file needs to be in

locale/\*code\*/LC\\_MESSAGES/phpmyadmin.mo.

.. config:option:: $cfg['DefaultConnectionCollation']

:type: string

:default: ``'utf8\_general\_ci'``

Defines the default connection collation to use, if not user-defined.

See the `MySQL documentation for charsets

<http://dev.mysql.com/doc/mysql/en/charset-charsets.html>`\_

for list of possible values. This setting is

ignored when connected to Drizzle server.

.. config:option:: $cfg['Lang']

:type: string

:default: not set

Force language to use. The corresponding language file needs to be in

locale/\*code\*/LC\\_MESSAGES/phpmyadmin.mo.

.. config:option:: $cfg['FilterLanguages']

:type: string

:default: ``''``

Limit list of available languages to those matching the given regular

expression. For example if you want only Czech and English, you should

set filter to ``'^(cs|en)'``.

.. config:option:: $cfg['RecodingEngine']

:type: string

:default: ``'auto'``

You can select here which functions will be used for character set

conversion. Possible values are:

\* auto - automatically use available one (first is tested iconv, then

recode)

\* iconv - use iconv or libiconv functions

\* recode - use recode\\_string function

\* mb - use mbstring extension

\* none - disable encoding conversion

Enabled charset conversion activates a pull-down menu in the Export

and Import pages, to choose the character set when exporting a file.

The default value in this menu comes from

:config:option:`$cfg['Export']['charset']` and :config:option:`$cfg['Import']['charset']`.

.. config:option:: $cfg['IconvExtraParams']

:type: string

:default: ``'//TRANSLIT'``

Specify some parameters for iconv used in charset conversion. See

`iconv documentation <http://www.gnu.org/software/libiconv/documentati

on/libiconv/iconv\_open.3.html>`\_ for details. By default

``//TRANSLIT`` is used, so that invalid characters will be

transliterated.

.. config:option:: $cfg['AvailableCharsets']

:type: array

:default: array(...)

Available character sets for MySQL conversion. You can add your own

(any of supported by recode/iconv) or remove these which you don't

use. Character sets will be shown in same order as here listed, so if

you frequently use some of these move them to the top.

Web server settings

-------------------

.. config:option:: $cfg['OBGzip']

:type: string/boolean

:default: ``'auto'``

Defines whether to use GZip output buffering for increased speed in

:term:`HTTP` transfers. Set to

true/false for enabling/disabling. When set to 'auto' (string),

phpMyAdmin tries to enable output buffering and will automatically

disable it if your browser has some problems with buffering. IE6 with

a certain patch is known to cause data corruption when having enabled

buffering.

.. config:option:: $cfg['TrustedProxies']

:type: array

:default: array()

Lists proxies and HTTP headers which are trusted for

:config:option:`$cfg['Servers'][$i]['AllowDeny']['order']`. This list is by

default empty, you need to fill in some trusted proxy servers if you

want to use rules for IP addresses behind proxy.

The following example specifies that phpMyAdmin should trust a

HTTP\\_X\\_FORWARDED\\_FOR (``X -Forwarded-For``) header coming from the proxy

1.2.3.4:

.. code-block:: php

$cfg['TrustedProxies'] = array('1.2.3.4' => 'HTTP\_X\_FORWARDED\_FOR');

The :config:option:`$cfg['Servers'][$i]['AllowDeny']['rules']` directive uses the

client's IP address as usual.

.. config:option:: $cfg['GD2Available']

:type: string

:default: ``'auto'``

Specifies whether GD >= 2 is available. If yes it can be used for MIME

transformations. Possible values are:

\* auto - automatically detect

\* yes - GD 2 functions can be used

\* no - GD 2 function cannot be used

.. config:option:: $cfg['CheckConfigurationPermissions']

:type: boolean

:default: true

We normally check the permissions on the configuration file to ensure

it's not world writable. However, phpMyAdmin could be installed on a

NTFS filesystem mounted on a non-Windows server, in which case the

permissions seems wrong but in fact cannot be detected. In this case a

sysadmin would set this parameter to ``false``.

.. config:option:: $cfg['LinkLengthLimit']

:type: integer

:default: 1000

Limit for length of :term:`URL` in links. When length would be above this

limit, it is replaced by form with button. This is required as some web

servers (:term:`IIS`) have problems with long :term:`URL` .

.. config:option:: $cfg['CSPAllow']

:type: string

:default: ``''``

Additional string to include in allowed script and image sources in Content

Security Policy header.

This can be useful when you want to include some external JavaScript files

in :file:`config.footer.inc.php` or :file:`config.header.inc.php`, which

would be normally not allowed by Content Security Policy.

To allow some sites, just list them within the string:

.. code-block:: php

$cfg['CSPAllow'] = 'example.com example.net';

.. versionadded:: 4.0.4

.. config:option:: $cfg['DisableMultiTableMaintenance']

:type: boolean

:default: false

In the database Structure page, it's possible to mark some tables then

choose an operation like optimizing for many tables. This can slow

down a server; therefore, setting this to ``true`` prevents this kind

of multiple maintenance operation.

Theme settings

--------------

.. config:option:: $cfg['NaviWidth']

:type: integer

:default:

Navigation panel width in pixels. See

:file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['NaviBackground']

:type: string [CSS color for background]

:default:

.. config:option:: $cfg['MainBackground']

:type: string [CSS color for background]

:default:

The background styles used for both the frames. See

:file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['NaviPointerBackground']

:type: string [CSS color for background]

:default:

.. config:option:: $cfg['NaviPointerColor']

:type: string [CSS color]

:default:

The style used for the pointer in the navigation panel. See

:file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['Border']

:type: integer

:default:

The size of a table's border. See :file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['ThBackground']

:type: string [CSS color for background]

:default:

.. config:option:: $cfg['ThColor']

:type: string [CSS color]

:default:

The style used for table headers. See

:file:`themes/themename/layout.inc.php`.

.. \_cfg\_BgcolorOne:

.. config:option:: $cfg['BgOne']

:type: string [CSS color]

:default:

The color (HTML) #1 for table rows. See

:file:`themes/themename/layout.inc.php`.

.. \_cfg\_BgcolorTwo:

.. config:option:: $cfg['BgTwo']

:type: string [CSS color]

:default:

The color (HTML) #2 for table rows. See

:file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['BrowsePointerBackground']

:type: string [CSS color]

:default:

The background color used when hovering over a row in the Browse panel.

See :file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['BrowsePointerColor']

:type: string [CSS color]

:default:

The text color used when hovering over a row in the Browse panel.

Used when :config:option:`$cfg['BrowsePointerEnable']` is true.

See :file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['BrowseMarkerBackground']

:type: string [CSS color]

:default:

The background color used to highlight a row selected by checkbox in the Browse panel or

when a column is selected.

Used when :config:option:`$cfg['BrowsePointerEnable']` is true.

See :file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['BrowseMarkerColor']

:type: string [CSS color]

:default:

The color used when you visually mark a row or column in the Browse panel.

Rows can be marked by clicking the checkbox to the left of the row and columns can be

marked by clicking the column's header (outside of the header text).

See :file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['FontFamily']

:type: string

:default:

You put here a valid CSS font family value, for example ``arial, sans-

serif``. See :file:`themes/themename/layout.inc.php`.

.. config:option:: $cfg['FontFamilyFixed']

:type: string

:default:

You put here a valid CSS font family value, for example ``monospace``.

This one is used in textarea. See :file:`themes/themename/layout.inc.php`.

Design customization

--------------------

.. config:option:: $cfg['NavigationTreePointerEnable']

:type: boolean

:default: true

When set to true, hovering over an item in the navigation panel causes that item to be marked

(the background is highlighted).

.. config:option:: $cfg['BrowsePointerEnable']

:type: boolean

:default: true

When set to true, hovering over a row in the Browse page causes that row to be marked (the background

is highlighted).

.. config:option:: $cfg['BrowseMarkerEnable']

:type: boolean

:default: true

When set to true, a data row is marked (the background is highlighted) when the row is selected

with the checkbox.

.. config:option:: $cfg['LimitChars']

:type: integer

:default: 50

Maximum number of characters shown in any non-numeric field on browse

view. Can be turned off by a toggle button on the browse page.

.. config:option:: $cfg['RowActionLinks']

:type: string

:default: ``'left'``

Defines the place where table row links (Edit, Copy, Delete) would be

put when tables contents are displayed (you may have them displayed at

the left side, right side, both sides or nowhere).

.. config:option:: $cfg['RowActionLinksWithoutUnique']

:type: boolean

:default: false

Defines whether to show row links (Edit, Copy, Delete) and checkboxes

for multiple row operations even when the selection does not have a unique key.

Using row actions in the absence of a unique key may result in different/more

rows being affected since there is no guaranteed way to select the exact row(s).

.. config:option:: $cfg['RememberSorting']

:type: boolean

:default: true

If enabled, remember the sorting of each table when browsing them.

.. config:option:: $cfg['TablePrimaryKeyOrder']

:type: string

:default: ``'NONE'``

This defines the default sort order for the tables, having a primary key,

when there is no sort order defines externally.

Acceptable values : ['NONE', 'ASC', 'DESC']

.. config:option:: $cfg['ShowBrowseComments']

:type: boolean

:default: true

.. config:option:: $cfg['ShowPropertyComments']

:type: boolean

:default: true

By setting the corresponding variable to ``true`` you can enable the

display of column comments in Browse or Property display. In browse

mode, the comments are shown inside the header. In property mode,

comments are displayed using a CSS-formatted dashed-line below the

name of the column. The comment is shown as a tool-tip for that

column.

Text fields

-----------

.. config:option:: $cfg['CharEditing']

:type: string

:default: ``'input'``

Defines which type of editing controls should be used for CHAR and

VARCHAR columns. Applies to data editing and also to the default values

in structure editing. Possible values are:

\* input - this allows to limit size of text to size of columns in MySQL,

but has problems with newlines in columns

\* textarea - no problems with newlines in columns, but also no length

limitations

.. config:option:: $cfg['MinSizeForInputField']

:type: integer

:default: 4

Defines the minimum size for input fields generated for CHAR and

VARCHAR columns.

.. config:option:: $cfg['MaxSizeForInputField']

:type: integer

:default: 60

Defines the maximum size for input fields generated for CHAR and

VARCHAR columns.

.. config:option:: $cfg['TextareaCols']

:type: integer

:default: 40

.. config:option:: $cfg['TextareaRows']

:type: integer

:default: 15

.. config:option:: $cfg['CharTextareaCols']

:type: integer

:default: 40

.. config:option:: $cfg['CharTextareaRows']

:type: integer

:default: 2

Number of columns and rows for the textareas. This value will be

emphasized (\\*2) for :term:`SQL` query

textareas and (\\*1.25) for :term:`SQL`

textareas inside the query window.

The Char\\* values are used for CHAR

and VARCHAR editing (if configured via :config:option:`$cfg['CharEditing']`).

.. config:option:: $cfg['LongtextDoubleTextarea']

:type: boolean

:default: true

Defines whether textarea for LONGTEXT columns should have double size.

.. config:option:: $cfg['TextareaAutoSelect']

:type: boolean

:default: false

Defines if the whole textarea of the query box will be selected on

click.

.. config:option:: $cfg['EnableAutocompleteForTablesAndColumns']

:type: boolean

:default: true

Whether to enable autocomplete for table and column names in any

SQL query box.

SQL query box settings

----------------------

.. config:option:: $cfg['SQLQuery']['Edit']

:type: boolean

:default: true

Whether to display an edit link to change a query in any SQL Query

box.

.. config:option:: $cfg['SQLQuery']['Explain']

:type: boolean

:default: true

Whether to display a link to explain a SELECT query in any SQL Query

box.

.. config:option:: $cfg['SQLQuery']['ShowAsPHP']

:type: boolean

:default: true

Whether to display a link to wrap a query in PHP code in any SQL Query

box.

.. config:option:: $cfg['SQLQuery']['Refresh']

:type: boolean

:default: true

Whether to display a link to refresh a query in any SQL Query box.

Web server upload/save/import directories

-----------------------------------------

.. config:option:: $cfg['UploadDir']

:type: string

:default: ``''``

The name of the directory where :term:`SQL` files have been uploaded by

other means than phpMyAdmin (for example, ftp). Those files are available

under a drop-down box when you click the database or table name, then the

Import tab.

If

you want different directory for each user, %u will be replaced with

username.

Please note that the file names must have the suffix ".sql"

(or ".sql.bz2" or ".sql.gz" if support for compressed formats is

enabled).

This feature is useful when your file is too big to be

uploaded via :term:`HTTP`, or when file

uploads are disabled in PHP.

.. note::

If PHP is running in safe mode, this directory must be owned by the same

user as the owner of the phpMyAdmin scripts. See also :ref:`faq1\_16` for

alternatives.

.. config:option:: $cfg['SaveDir']

:type: string

:default: ``''``

The name of the directory where dumps can be saved.

If you want different directory for each user, %u will be replaced with

username.

Please note that the directory must exist and has to be writable for

the user running webserver.

.. note::

If PHP is running in safe mode, this directory must be owned by the same

user as the owner of the phpMyAdmin scripts.

.. config:option:: $cfg['TempDir']

:type: string

:default: ``''``

The name of the directory where temporary files can be stored.

This is needed for importing ESRI Shapefiles, see :ref:`faq6\_30` and to

work around limitations of ``open\_basedir`` for uploaded files, see

:ref:`faq1\_11`.

If the directory where phpMyAdmin is installed is

subject to an ``open\_basedir`` restriction, you need to create a

temporary directory in some directory accessible by the web server.

However for security reasons, this directory should be outside the

tree published by webserver. If you cannot avoid having this directory

published by webserver, place at least an empty :file:`index.html` file

there, so that directory listing is not possible.

This directory should have as strict permissions as possible as the only

user required to access this directory is the one who runs the webserver.

If you have root privileges, simply make this user owner of this directory

and make it accessible only by it:

.. code-block:: sh

chown www-data:www-data tmp

chmod 700 tmp

If you cannot change owner of the directory, you can achieve a similar

setup using :term:`ACL`:

.. code-block:: sh

chmod 700 tmp

setfacl -m "g:www-data:rwx" tmp

setfacl -d -m "g:www-data:rwx" tmp

If neither of above works for you, you can still make the directory

:command:`chmod 777`, but it might impose risk of other users on system

reading and writing data in this directory.

Various display setting

-----------------------

.. config:option:: $cfg['RepeatCells']

:type: integer

:default: 100

Repeat the headers every X cells, or 0 to deactivate.

.. config:option:: $cfg['QueryHistoryDB']

:type: boolean

:default: false

.. config:option:: $cfg['QueryHistoryMax']

:type: integer

:default: 25

If :config:option:`$cfg['QueryHistoryDB']` is set to ``true``, all your

Queries are logged to a table, which has to be created by you (see

:config:option:`$cfg['Servers'][$i]['history']`). If set to false, all your

queries will be appended to the form, but only as long as your window is

opened they remain saved.

When using the JavaScript based query window, it will always get updated

when you click on a new table/db to browse and will focus if you click on

:guilabel:`Edit SQL` after using a query. You can suppress updating the

query window by checking the box :guilabel:`Do not overwrite this query

from outside the window` below the query textarea. Then you can browse

tables/databases in the background without losing the contents of the

textarea, so this is especially useful when composing a query with tables

you first have to look in. The checkbox will get automatically checked

whenever you change the contents of the textarea. Please uncheck the button

whenever you definitely want the query window to get updated even though

you have made alterations.

If :config:option:`$cfg['QueryHistoryDB']` is set to ``true`` you can

specify the amount of saved history items using

:config:option:`$cfg['QueryHistoryMax']`.

.. config:option:: $cfg['BrowseMIME']

:type: boolean

:default: true

Enable :ref:`transformations`.

.. config:option:: $cfg['MaxExactCount']

:type: integer

:default: 500000

For InnoDB tables, determines for how large tables phpMyAdmin should

get the exact row count using ``SELECT COUNT``. If the approximate row

count as returned by ``SHOW TABLE STATUS`` is smaller than this value,

``SELECT COUNT`` will be used, otherwise the approximate count will be

used.

.. config:option:: $cfg['MaxExactCountViews']

:type: integer

:default: 0

For VIEWs, since obtaining the exact count could have an impact on

performance, this value is the maximum to be displayed, using a

``SELECT COUNT ... LIMIT``. Setting this to 0 bypasses any row

counting.

.. config:option:: $cfg['NaturalOrder']

:type: boolean

:default: true

Sorts database and table names according to natural order (for

example, t1, t2, t10). Currently implemented in the navigation panel

and in Database view, for the table list.

.. config:option:: $cfg['InitialSlidersState']

:type: string

:default: ``'closed'``

If set to ``'closed'``, the visual sliders are initially in a closed

state. A value of ``'open'`` does the reverse. To completely disable

all visual sliders, use ``'disabled'``.

.. config:option:: $cfg['UserprefsDisallow']

:type: array

:default: array()

Contains names of configuration options (keys in ``$cfg`` array) that

users can't set through user preferences. For possible values, refer

to :file:`libraries/config/user\_preferences.forms.php`.

.. config:option:: $cfg['UserprefsDeveloperTab']

:type: boolean

:default: false

Activates in the user preferences a tab containing options for

developers of phpMyAdmin.

Page titles

-----------

.. config:option:: $cfg['TitleTable']

:type: string

:default: ``'@HTTP\_HOST@ / @VSERVER@ / @DATABASE@ / @TABLE@ | @PHPMYADMIN@'``

.. config:option:: $cfg['TitleDatabase']

:type: string

:default: ``'@HTTP\_HOST@ / @VSERVER@ / @DATABASE@ | @PHPMYADMIN@'``

.. config:option:: $cfg['TitleServer']

:type: string

:default: ``'@HTTP\_HOST@ / @VSERVER@ | @PHPMYADMIN@'``

.. config:option:: $cfg['TitleDefault']

:type: string

:default: ``'@HTTP\_HOST@ | @PHPMYADMIN@'``

Allows you to specify window's title bar. You can use :ref:`faq6\_27`.

Theme manager settings

----------------------

.. config:option:: $cfg['ThemePath']

:type: string

:default: ``'./themes'``

If theme manager is active, use this as the path of the subdirectory

containing all the themes.

.. config:option:: $cfg['ThemeManager']

:type: boolean

:default: true

Enables user-selectable themes. See :ref:`faqthemes`.

.. config:option:: $cfg['ThemeDefault']

:type: string

:default: ``'pmahomme'``

The default theme (a subdirectory under :config:option:`$cfg['ThemePath']`).

.. config:option:: $cfg['ThemePerServer']

:type: boolean

:default: false

Whether to allow different theme for each server.

Default queries

---------------

.. config:option:: $cfg['DefaultQueryTable']

:type: string

:default: ``'SELECT \* FROM @TABLE@ WHERE 1'``

.. config:option:: $cfg['DefaultQueryDatabase']

:type: string

:default: ``''``

Default queries that will be displayed in query boxes when user didn't

specify any. You can use standard :ref:`faq6\_27`.

MySQL settings

--------------

.. config:option:: $cfg['DefaultFunctions']

:type: array

:default: array(...)

Functions selected by default when inserting/changing row, Functions

are defined for meta types as (FUNC\\_NUMBER, FUNC\\_DATE, FUNC\\_CHAR,

FUNC\\_SPATIAL, FUNC\\_UUID) and for ``first\_timestamp``, which is used

for first timestamp column in table.

Developer

---------

.. warning::

These settings might have huge effect on performance or security.

.. config:option:: $cfg['DBG']

:type: array

:default: array(...)

.. config:option:: $cfg['DBG']['sql']

:type: boolean

:default: false

Enable logging queries and execution times to be

displayed in the console's Debug SQL tab.

.. config:option:: $cfg['DBG']['demo']

:type: boolean

:default: false

Enable to let server present itself as demo server.

This is used for <http://demo.phpmyadmin.net/>.