BROWSER INTERFACE

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Table of Content

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Chapter 1: Picture reference

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Chapter 2: Definitions and abbreviations

[APP]	"Application" is a Word Wide Web server-based application.
[DB]	"Application Data Base" is the place where [APP] data is stored. The layout of the [RECORD]s (and the location where the records are stored) depends on [APP] internal architecture.
[CONNECTOR]	The "Data base connection handler" is used by [SERVER] to access records in [DB].
[CLIENT]	Client device on which user initiates synchronization session. The [CLIENT] may be a cell phone or personnel digital assistant (PDA). On most [CLIENT], the synchronization capabilities are build-in; on other [CLIENT] it may be required to install special client software.
syncgw	**syncgw** server acts as a gateway between [CLIENT] and [DB]. The server provides data conversion utilities and support of a couple of data protocols enabling communication with [CLIENT].
[CONFIG]	**syncgw** configuration is stored in configuration file `syncgw/config.ini.php`. This file edited using the [BROWSER] interface.
[BROWSER]	The standalone **syncgw** "Web User interface" is available at `http://[yourdomain]/sync.php`.
[DATASTORE]	"Data Store" contains all related data to a specific data store (e.g. all contact data). Depending on which [CONNECTOR] is used, the data may be physically stored in MySQL tables, in flat files or somewhere else (e.g. in an Oracle data base). Data stored may also contain **syncgw** internal data records (e.g. trace data) or shared data records (e.g. attachment data).
[USERDATA]	These are all [DATASTORE]s in which user data is stored (e.g. contact data, calendar events).
[GROUP]	All records are stored in groups. These may be user defined. If no group is selected during synchronization, **syncgw** uses a special group called "Default" in each [DATASTORE] to synchronize. Please note this group cannot be deleted and is always automatically (re-) created. During [SYNCML] synchronization, only the "Default" group is used. Other groups may be selected using a [WEBDAV] synchronization.
[RECORD]	A "Data Record" contains an indefinite number of [FIELD]s. In case of e.g. a contact [RECORD], there may be one name with multiple telephone numbers assigned and multiple location information such as work and private address.
[FIELD]	A "Data field" is one piece of information store in a [RECORD]. In case of a contact [RECORD], each telephone number is a [FIELD].
[GUID]	The "Global Unique Identified" is the primary key used by **syncgw** to access a specific [RECORD] in a specific [DATASTORE] for a specific user.
[LUID]	The "Local Unique Identifier" is the primary key used by the [CLIENT] to access a specific [RECORD] in a specific [DATASTORE] for a specific user.

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Record mapping	During access to external [DATASTORE] all internal [FIELD] needs to be mapped to external [FIELD]. This process is called “Record mapping”. This transformation is performed based on an XML mapping table.
Character Encoding	[SERVER] use internally the UTF-8-character set. During conversation with [CLIENT], all user data is automatically converted to the character set supported by [CLIENT]. If you want to know more about the supported character sets, please open [BROWSER] and select “Check status”. Please search for “Encoding handler”. Below the header line all supported character sets are listed.
[WAP]	The “Wireless Application Protocol” used for transferring data across 3G networks.
[SYNCML]	“Synchronization Markup Language” is a standard protocol for synchronizing client devices with server application.
[WBXML]	“WAP Binary XML” is a binary representation of [SYNCML] protocol which is used to synchronize smaller data packages during synchronization.
[WEBDAV]	“Web-based Distributed Authoring and Versioning” is a set of methods based on the Hypertext Transfer Protocol (HTTP) that facilitates collaboration between users in editing and managing documents and files stored on Word Wide Web servers.
[MS-EAS]	MicroSoft Exchange ActiveSync

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Chapter 3: Introduction

[BROWSER] is the web interface used for configuring **syncogw** and managing data stores. Please enter `http://[yourdomain]/sync.php` in the URL of your favorite browser to open [BROWSER].

Chapter 4: Login to [BROWSER]

1.1 Define a [BROWSER] administrator password

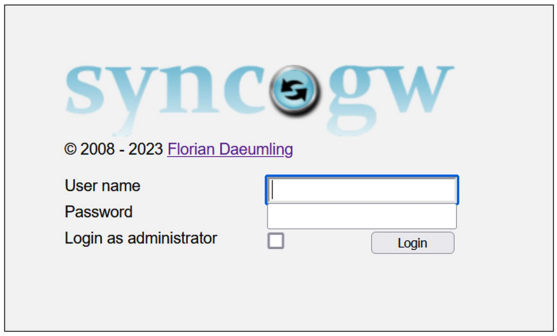
Picture 1: Password definition

The first time you start [BROWSER] you are prompted for an administrator password. You may enter any pass phrase you want. The password is stored in [CONFIG]. The password itself is encrypted and cannot be recovered.

If you encounter the need to change the password, there is only one way to succeed. Please edit [CONFIG] and delete the line with the `AdminPassword` parameter. On next execution of [BROWSER], you will be automatically prompted for a new password.

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

1.2 Login to [BROWSER]



Picture 2: Login panel

If you click on “Login as administrator” checkbox “User name” is grayed out and you may login as administrator using the administrator password (there is no administrator user ID).

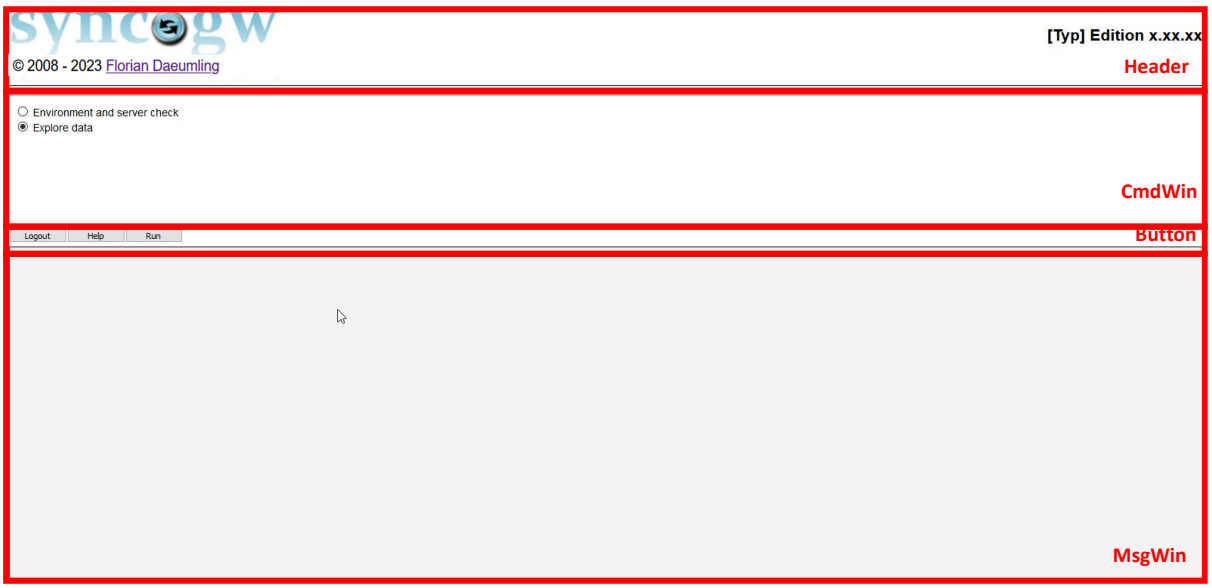
All other user may login using their known “User name” and “Password”. Please note user may only login after the first successful synchronization has been performed in depended of their login status in [APP].

As soon as you’re logged in, you may log out using the Logout button. We recommend always using the Logout button. Please keep in mind, if you close browser window and leaves the computer without logging off, another user may open [BROWSER] from browser history and is not required to authenticate.

Depending on your login data, user will see a different number of available commands.

Chapter 5: Window sections

The [BROWSER] window is split in multiple sections.



Picture 3: [BROWSER] layout

In **Header** section `[Typ]` is replaced either by **Standard** or **Professional**; **x.xx.xx** is replaced by **sync•gw** version number.

TopWin contains all available commands.

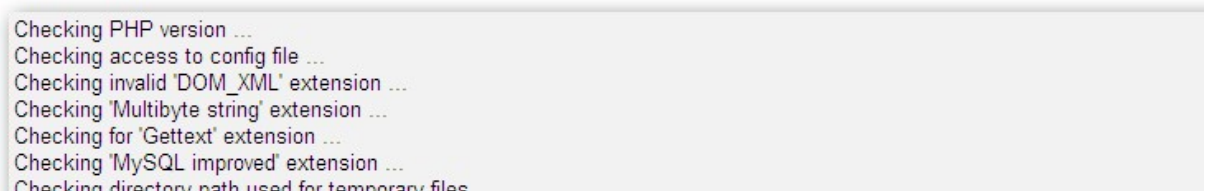
Button section contains all available command buttons. Please note, the buttons are context sensitive – depending on which command you are executing, the number of available command buttons may vary.

MsgWin contains the output of the last command executed.

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Chapter 6: Check status

This command is available to check PHP environment and sync•gw status. If sync•gw is not yet configured and you select the “Check status” command and hit the Run button, you will receive the error message shown below.



Picture 4: Status output

We recommend using the “Check status” command each time you update sync•gw software. This will ensure sync•gw is properly configured and perform any automatic upgrade activity if required.

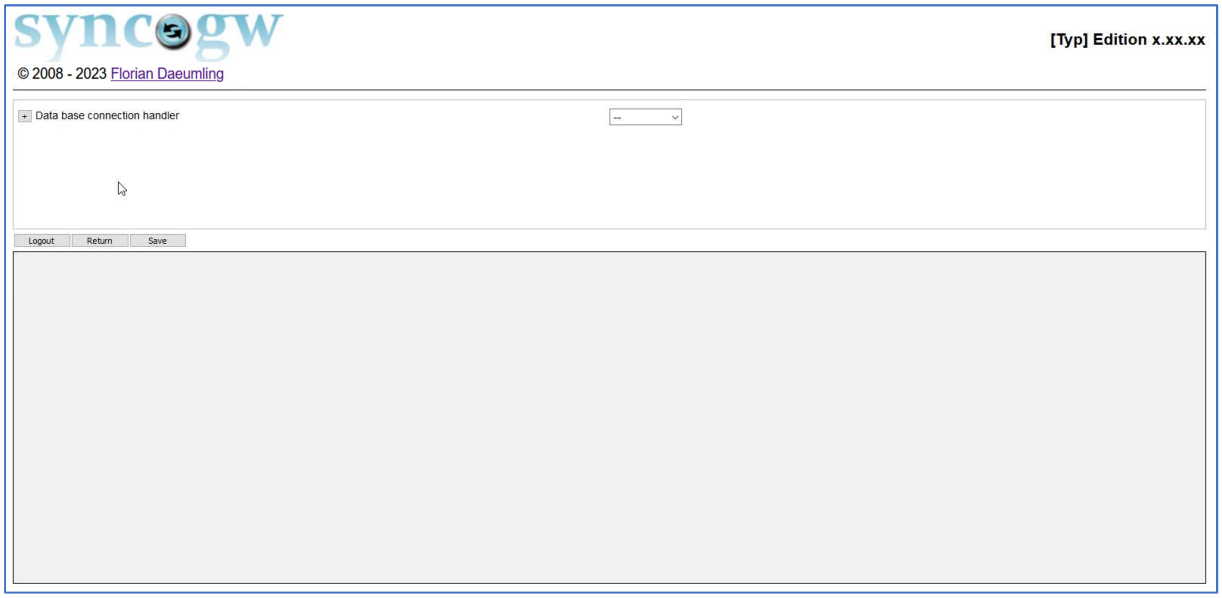
If everything is ok, no error message will be shown and at the bottom of the window you should see a message sync•gw is ready for synchronizing.

Picture 5: Overall status

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Chapter 8: Configure syncgw

If you logged in as administrator, you you're able to modify all available **syncgw** configuration parameters. For a complete list of available configuration parameter please refer to "Appendix A: Configuration parameter".



Picture 6: Configuration panel

In front of each parameter there is a twisty available to open/close a help section. Clicking on the ⊞ button opens a help text below the parameter including a description of available configuration option values. Clicking on ⊟ will close the help section again.

If you've installed **Professional Edition**, an additional informational message will appear in the **Header** section. This information shows how much memory is used by [BROWSER] script calling the PHP function `memory_get_usage()` and the peak usage calling the `memory_get_peak_usage()` function.

Chapter 9: View syncgw log

If you've specified a log file in configuration panel, the additional command "Show sync•gw log file" is available to display the internal log file.

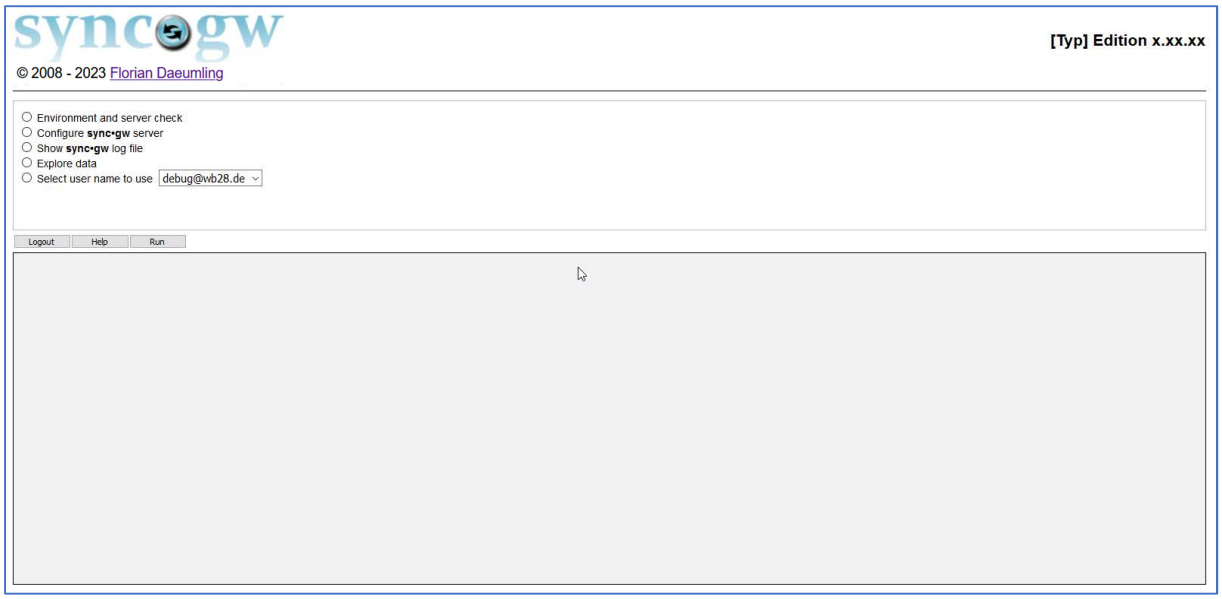
Picture 7: sync•gw log file

The log display is dynamic. All messages previously written to log file (and all new messages) appears in **MsgWin** in log file panel. This can be used to monitor sync•gw server execution.

With the Clear button, you can delete existing log file content.

If you click on the Stop button, the dynamic update is suspended, and a notification message is shown on upper right-hand side; then the button text is changes to Start. Clicking on Start enables dynamic updates again.

Chapter 10: Select user



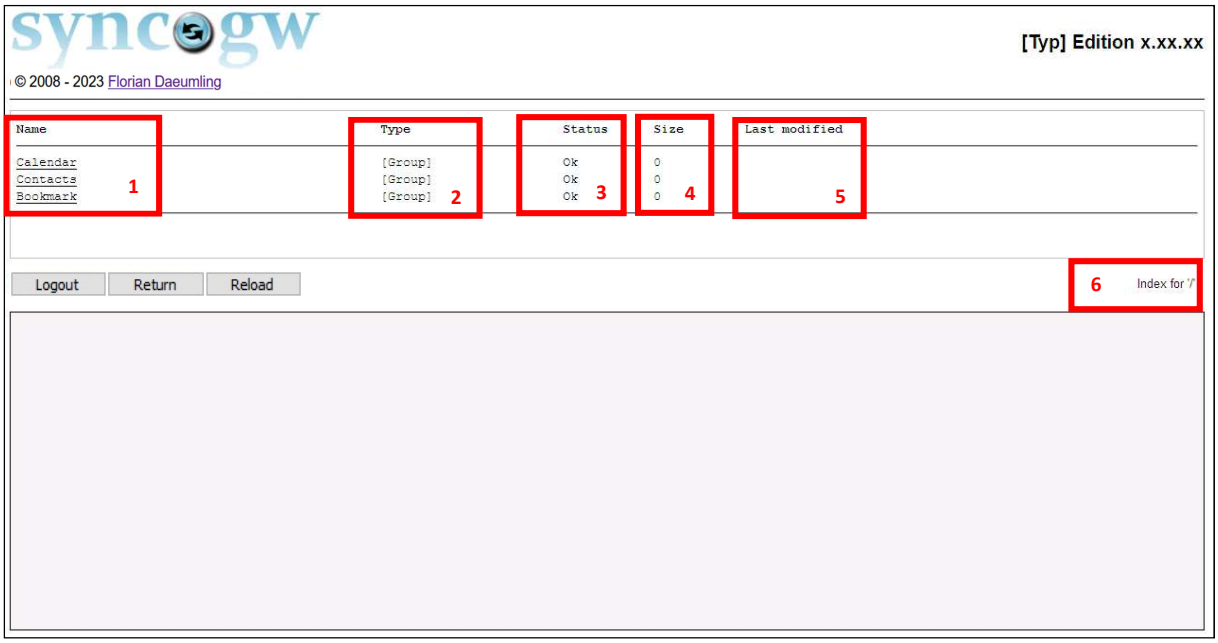
Picture 8: Selecting user

If you're logged in as administrator, you may access [USERDATA] only switching to specific user. Please select the user from drop down and click on "Select user name to use". Then you may explore specific [USERDATA].

Chapter 11: Explore data

1.1 [DATASTORE] level #1

The command “Explore data” is available as soon as a [CONNECTOR] in configuration panel is selected. In explorer panel you may examine any syncgw [RECORDS] available in [DATASTORE].



Picture 9: [DATASTORE] level #1 view

Above you see the initial display of the explorer panel. Depending on the configured [DATASTORES] you will see more or less [DATASTORE].

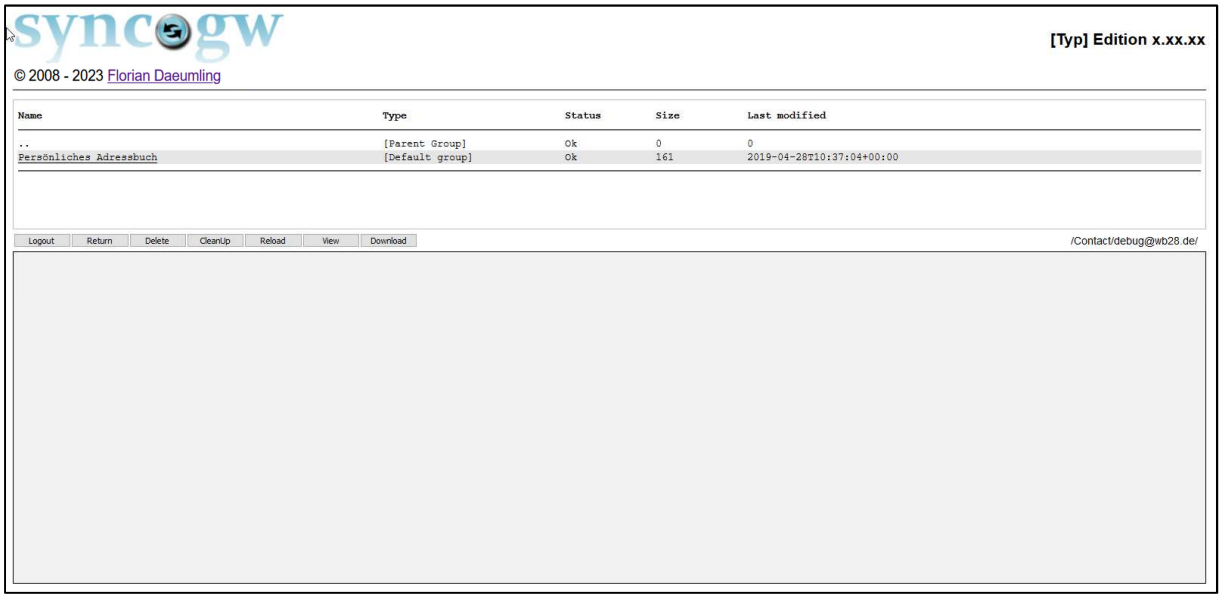
1 List all records or groups on this level
2 Is either “Group”, “Default Group”, “Parent Group” or “Record”
3 [RECORD] status is “Ok” (synchronized), “Add” (to be send to client device), “Delete” (to be deleted on client device) or “Replace” (to be replaced on client device)
4 Size of record or group
5 Last time of modification
6 Index level you explore data

Use Reload button to reload the current [DATASTORE]. This is usefully to catch any updates on [DATASTORE] performed while you’re inspecting the content (e.g. if a user performs a new synchronization session while you’re viewing data).

If you’re logged in as administrator, all internal [DATASTORE] will also be shown. These are:

“Session”	Session data store used for storage of internal session related data.
“Trace”	Trace data store used for storage of trace data.
“Device”	Device information storage.
“User”	User information storage.
“Attachments”	Attachment data storage.

1.2 [DATASTORE] level #2



Picture 10: [DATASTORE] level #2 view

If you select **"**Contacts" [DATASTORE], you will be presented with n different view.

There is a new button available. If you click Return button, you will leave explorer view and return to main menu. If you Delete click button, the selected record will be deleted; button CleanUp will delete all record in data store. The View button is used to show the [DATASTORE] record. The Download button is useful, if you want to download a single record from [DATASTORE].

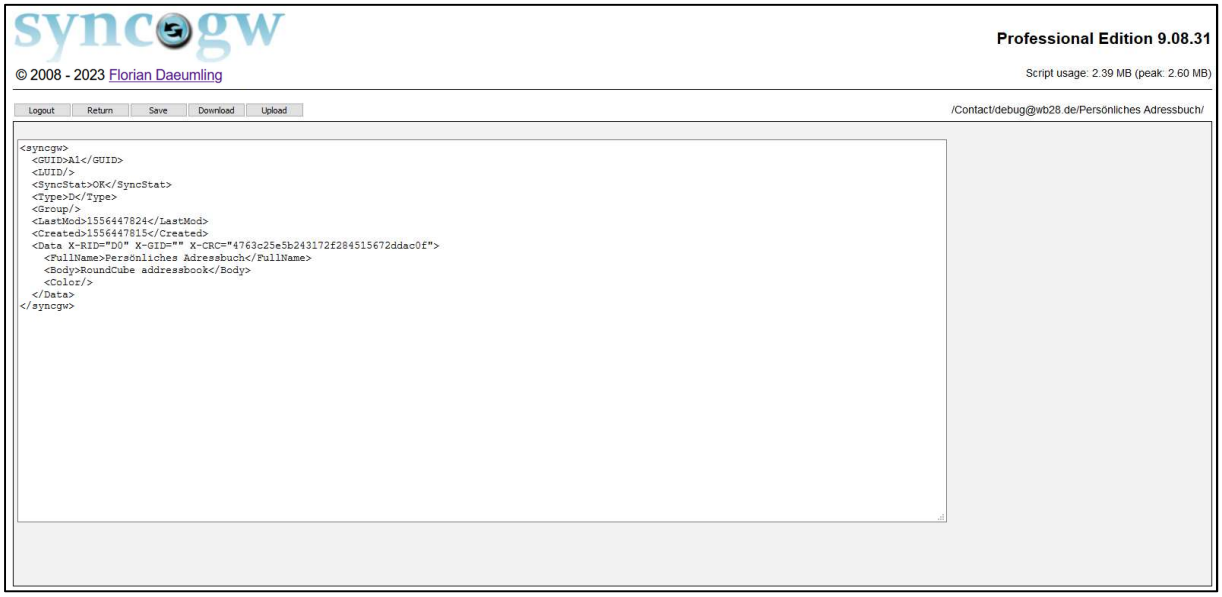
If you're in "User" [DATASTORE] and either delete a user (or click on CleanUp button) in "User" data store, all user related record will be deleted (including all related records in "contacts", "calendar" and other data stores.

Clicking on ".." returns you to level #1.

1.3 Additional command buttons in Professional Edition

If you've purchased the **Professional Edition** you have the following addition command buttons available.

If you select a record and click on Edit button you may update the record.



Picture 11: Edit data record

Use either the Return button to abort edit or save your changes clicking on the Save button.

With Upload button you may upload a record to syncgw internal [DATASTORE].

In [USERDATA] additional button Sync is available. For more information about this topic please read our "Developer Guide" documentation available in our download section.

With Rename button you're allowed to rename internal [DATASTORE] record.

1.4 User [DATASTORE]

An additional button Stats is available to show some statistics about all users available in sync•gw:

User statistics

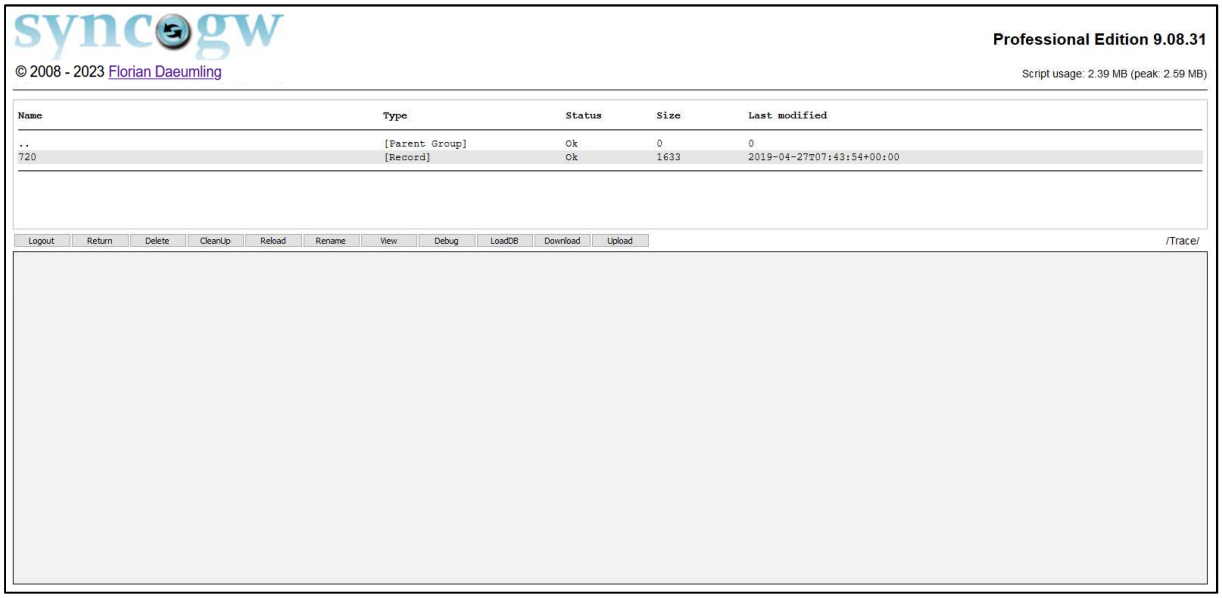
User name	debug@
User ID	1
Last login	2019-04-27T07:43:54+00:00
Number of logins	20
Active client device name	Dbg-NineBC6F50B39720

Picture 12: User statistics

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

1.5 Trace [DATASTORE]

In “Trace” [DATASTORE] some additional command buttons are available in **Professional Edition**.



Picture 13: Trace [DATASTORE]

During creation of a trace record **sync•gw** saves the initial status of the internal [DATASTORE] and the [DB] and then captures all send and received information exchanged with [CLIENT].

To load the trace environment for the “Debug user id” specified in configuration panel, please select a trace record and click on LoadDB button. Then all available external and internal [RECORD] are restored from trace record.

You are now able to inspect the records in [APP] and the internal records as they appear **before** synchronization takes place and any updated are performed.

Using the Debug button will execute synchronization session, showing all data created as during a normal synchronization. During debugging [BROWSER] emulates the synchronization session between [CLIENT] and **sync•gw**.

Please note in opposite to the View button (which only shows the captured trace data during synchronization session) **sync•gw** processes all “received” data from [CLIENT] and creates new output.

You may use this function to check your [CONNECTOR] processing or to figure out any session misbehavior. Our support team uses trace records to analyze any misbehavior and tests updates on **sync•gw** code received from our development department

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Appendix A: Configuration parameter

1.1 General configuration parameter

This chapter contains a detailed description of all available configuration options used in [CONFIG] file.

- Data base connection handler (Parameter name: `Database`)

Select from list which data base handler **sync•gw** should use. If you can't select a handler name, a connection is already established. To drop the connection, use the "Drop" button.

Depending on selected "*Data base connection handler*" there are some additional parameters to specify.

- Enabled data store (Parameter name: `Datastores`)

Specify which data stores you want to be enabled and available for synchronization with remote devices. If a handler is not selectable, you have either not purchased the handler modules or your application back end handler do not support this type of data store.

- Administrator password (Parameter name: `AdminPassword`)

Please enter new **sync•gw** administrator password.

- Language (Parameter name: `Language`)

sync•gw server supports gettext based native language support - all message can be translated to any required language.

If your required language is not in selection list, you may create your own translation file. For this purpose, we recommend using one of translation tools available (e.g. KBabel, GTranslator or POEdit).

We would greatly appreciate if you share the translation file within our community. Please send the resulting .PO file to nls@syncgw.com and we will make it available for all users in our download section at syncgw.com.

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Capture PHP error (Parameter name: `CronJob`)

By default **sync•gw** is able to catch all PHP warning and notices. Setting this option to **Yes** enables **sync•gw** additionally to capture all PHP fatal errors in the log file specified above. Please note **sync•gw** will override locally some PHP.ini settings.

Use CRON job (Parameter name: `PHPError`)

By default **sync•gw** is handling record expiration internally. This solution may have impact on synchronization performance. We recommend setup your own CRON job. For this purpose please call **sync.php?cleanup** at least every hour. If you're using PLESK, you may call **sync.php** as script with parameter **cleanup**.

Log file name (Parameter name: `LogFile`)

Specify where to store error, warning and informational messages.

Off
Turn off any logging.

SysLog
Log messages to system log file.

<name>
Write log messages to file. You may specify either a relative file name prefix (e.g. "../logs/syncgw-log") or an absolute path (e.g. "/var/logs/syncgw")

Logging level (Parameter name: `LogLevel`)

sync•gw server may write errors, warnings and other messages to log file. Depending on your setting, your log file will use more or less disk space.

Error
Show errors which **sync•gw** either cannot handle or were unexpected (will always be logged).

Warn
Show additional warnings which **sync•gw** can cover.

Info
Show additional informational messages.

Application
Additional application processing messages.

Debug
More detailed processing messages.

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

- Log file expiration (Parameter name: `LogFileExp`)

Specify how many log files should be kept before the eldest file will be deleted.

- Debug user (Parameter name: `DebugUser`)

This user is used to access the internal and external data bases (e.g. during debugging traces). Debug user must be authorized to access internal and external data base. To disable debugging, please leave this field empty. Please note, a couple of additional functions in "Explore data" panel will only be available, if you have specified a debug user.

- Debug user password (Parameter name: `DebugUserPassword`)

Password for debug user.

- Trace (Parameter name: `Trace`)

Trace data is used to enable debugging of any misbehavior of **sync•gw** server. If you encounter any problems, we need such a trace to analyze the situation. Available options:

On
Activate tracing for all users.

<IP>
Enable tracing for sprecific IP address.

<UserName>
Enable tracing only for user <UserName>.

Off
Disable tracing for all users.

- Trace directory (Parameter name: `TraceDir`)

Specify where to store trace files. You may specify either a relative directory name prefix (e.g. "../traces") or an absolute path (e.g. "/var/traces").

- Trace file expiration (in hours) (Parameter name: `TraceExp`)

After the given number of hours **sync•gw** automatically removes expired trace files from trace directory. If you want to disable automatic file deletion, please enter a value of **0**.

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

Session timeout (in seconds) (Parameter name: `TraceExp`)

After the given number of hours **sync•gw** automatically removes expired trace files from trace directory. If you want to disable automatic file deletion, please enter a value of **0**.

Session record expiration (in hours) (Parameter name: `SessionExp`)

sync•gw stores record for managing synchronization sessions records in internal data stores. After the given number of hours **sync•gw** automatically removes records expired from internal data stores. If you want to disable automatic record deletion, please enter a value of **0**.

Maximum object size in bytes for DAV synchronization (Parameter name: `MaxObjectSize`)

This is the maximum size object **sync•gw** server accepts (in bytes, "KB", "MB" or "GB") for DAV synchronization.

Please note the size is limited by two factors:

- The PHP maximum excution time.
- The PHP memory_limit size.

We highly recommend if you want to use a bigger value, you should make some testing before taking over value over to production system.

Default: 1000 KB

ActiveSync Heartbeat window (in seconds) (Parameter name: `PingSleep`)

Using ActiveSync protocol, devices send a request to **sync•gw** server asking server to check for changes on server. If a change is recognized in this time window, device is notified immediately. If no changes could be notified, **sync•gw** server will send a notification after the heartbeat has expired. You can override the heartbeat client suggests to lower traffic between server and device. This parameter specifies how many seconds **sync•gw** server will check for changes before client is notified nothing has changed.

ActiveSync Sleep time (in seconds) (Parameter name: `HeartBeat`)

Within the heartbeat window, **sync•gw** will not constantly check for changes. This parameter specifies how many seconds **sync•gw** will sleep before checking for changes.

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)

END OF DOCUMENT

Copyright © 2008 - 2022 Florian Däumling, Germany (https://github.com/Toteph42)