Contents

1 NAME

OpenSLX::ConfigDB - the configuration database API class for OpenSLX

2 SYNOPSIS

```
use OpenSLX::ConfigDB;

openslxInit();

my $openslxDB = OpenSLX::ConfigDB->new();
$openslxDB->connect();

# fetch a client by name:
my $defaultClient = $openslxDB->fetchClientByFilter({'name' => '<<<default>>>'})

# fetch all systems:
my @systems = $openslxDB->fetchSystemByFilter();
```

3 DESCRIPTION

This class defines the OpenSLX API to the config database (the data layer to the outside world).

The ConfigDB interface contains of five different parts:

- basic methods (connection handling)

- data access methods (getting data)

- data manipulation methods (adding, removing and changing data)

- data aggregation methods (getting info about the resulting configurations after mixing individual client-, group- and system- configurations).

- suppport functions (useful helpers)

4 Special Concepts

`Filters`

A filter is a hash-ref defining the filter criteria to be applied to a database query. Each key of the filter corresponds to a DB column and the (hash-)value contains the respective column value.

[At a later stage, this will be improved to support a more structured approach to filtering (with boolean operators and hierarchical expressions)].

5 Methods

Basic Methods

`new()`

Returns an object representing a database handle to the config database.

`connect()`

Tries to establish a connection to the database specified via the db-... settings. The global configuration hash `%openslxConfig` contains further info about the requested connection. When implementing this method, you may have to look at the following entries in order to find out which database to connect to:

`$openslxConfig{'db-spec'}`

Full specification of database, a special string defining the precise database to connect to (this allows connecting to a database that requires specifications which aren't cared for by the existing `%config`-entries).

`$openslxConfig{'db-name'}`

The precise name of the database that should be connected (defaults to 'openslx').

`disconnect()`

Tears down the connection to the database and cleans up.

`startTransaction()`

Opens a database transaction - most useful if you want to make sure a couple of changes apply as a whole or not at all.

`commitTransaction()`

Commits a database transaction - so all changes done inside of this transaction will be applied to the database.

`rollbackTransaction()`

Revokes a database transaction - so all changes done inside of this transaction will be undone.

`cleanupAnyInconsistencies()`

Looks for any inconsistencies (stale references, references to non-existing plugins, ...) and removes them from the DB.

`synchronizeAttributesWithDB()`

Makes sure that all known attributes are referenced by the default system (and no unknown ones).

Additionally, all systems, groups and clients can be checked and get their stale attributes removed, too.

`_removeStaleSystemAttributes()`

Removes any stale attributes from every system.

`_removeStaleGroupAttributes()`

Removes any stale attributes from every group.

`_removeStaleClientAttributes()`

Removes any stale attributes from every client.

`_removeStaleVendorOSAttributes()`

Removes any stale attributes from every vendor-OS.

Data Access Methods

`getColumnsOfTable($tableName)`

Returns the names of the columns of the given table.

Param `tableName`

The name of the DB-table whose columns you'd like to retrieve.

Return Value

An array of column names.

`fetchVendorOSByFilter([%$filter], [$resultCols])`

Fetches and returns information about all vendor-OSes that match the given filter.

Param `filter`

A hash-ref containing the filter criteria that shall be applied - default is no filtering. See §**??** for more info.

Param `resultCols`

A string listing the columns that shall be returned - default is all columns.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchVendorOSByID(@$ids, [$resultCols])`

Fetches and returns information the vendor-OSes with the given IDs.

Param `ids`

An array of the vendor-OS-IDs you are interested in.

Param `resultCols`

A string listing the columns that shall be returned - default is all columns.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchInstalledPlugins($vendorOSID)`

Returns the names of all plugins that have been installed into the given vendor-OS.

Param `vendorOSID`

The id of the vendor-OS whose plugins you are interested in

Param `pluginName` **[Optional]**

The name of a specific plugin you are interested in

Return Value

An array with the plugin names.

`fetchExportByFilter([%$filter], [$resultCols])`

Fetches and returns information about all exports that match the given filter.

Param `filter`

A hash-ref containing the filter criteria that shall be applied - default is no filtering. See §**??** for more info.

Param `resultCols`

A string listing the columns that shall be returned - default is all columns.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchExportByID(@$ids, [$resultCols])`

Fetches and returns information the exports with the given IDs.

Param `ids`

An array of the export-IDs you are interested in.

Param `resultCols`

A string listing the columns that shall be returned - default is all columns.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchExportIDsOfVendorOS($id)`

Fetches the IDs of all exports that make use of the vendor-OS with the given ID.

Param `id`

ID of the vendor-OS whose exports shall be returned.

Return Value

An array of system-IDs.

`fetchGlobalInfo($id)`

Fetches the global info element specified by the given ID.

Param `id`

The name of the global info value you are interested in.

Return Value

The value of the requested global info.

`fetchSystemByFilter([%$filter], [$resultCols])`

Fetches and returns information about all systems that match the given filter.

Param `$filter`

A hash-ref containing the filter criteria that shall be applied - default is no filtering. See §**??** for more info.

Param `$resultCols` **[Optional]**

A comma-separated list of colunm names that shall be returned. If not defined, all available data must be returned.

Param `$attrFilter` **[Optional]**

A hash-ref containing the filter criteria that shall be applied against attributes.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchSystemByID(@$ids, [$resultCols])`

Fetches and returns information the systems with the given IDs.

Param `ids`

An array of the system-IDs you are interested in.

Param `resultCols`

A string listing the columns that shall be returned - default is all columns.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchSystemIDsOfExport($id)`

Fetches the IDs of all systems that make use of the export with the given ID.

Param `id`

ID of the export whose systems shall be returned.

Return Value

An array of system-IDs.

`fetchSystemIDsOfClient($id)`

Fetches the IDs of all systems that are used by the client with the given ID.

Param `id`

ID of the client whose systems shall be returned.

Return Value

An array of system-IDs.

`fetchSystemIDsOfGroup($id)`

Fetches the IDs of all systems that are part of the group with the given ID.

Param `id`

ID of the group whose systems shall be returned.

Return Value

An array of system-IDs.

`fetchClientByFilter([%$filter], [$resultCols])`

Fetches and returns information about all clients that match the given filter.

Param `$filter`

A hash-ref containing the filter criteria that shall be applied - default is no filtering. See §**??** for more info.

Param `$resultCols` **[Optional]**

A comma-separated list of colunm names that shall be returned. If not defined, all available data must be returned.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchClientByID(@$ids, [$resultCols])`

Fetches and returns information the clients with the given IDs.

Param `ids`

An array of the client-IDs you are interested in.

Param `resultCols`

A string listing the columns that shall be returned - default is all columns.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchClientIDsOfSystem($id)`

Fetches the IDs of all clients that make use of the system with the given ID.

Param `id`

ID of the system whose clients shall be returned.

Return Value

An array of client-IDs.

`fetchClientIDsOfGroup($id)`

Fetches the IDs of all clients that are part of the group with the given ID.

Param `id`

ID of the group whose clients shall be returned.

Return Value

An array of client-IDs.

`fetchGroupByFilter([%$filter], [$resultCols])`

Fetches and returns information about all groups that match the given filter.

Param `$filter`

A hash-ref containing the filter criteria that shall be applied - default is no filtering. See §**??** for more info.

Param `$resultCols` **[Optional]**

A comma-separated list of colunm names that shall be returned. If not defined, all available data must be returned.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchGroupByID(@$ids, [$resultCols])`

Fetches and returns information the groups with the given IDs.

Param `ids`

An array of the group-IDs you are interested in.

Param `resultCols`

A string listing the columns that shall be returned - default is all columns.

Return Value

An array of hash-refs containing the resulting data rows.

`fetchGroupIDsOfSystem($id)`

Fetches the IDs of all groups that contain the system with the given ID.

Param `id`

ID of the system whose groups shall be returned.

Return Value

An array of client-IDs.

`fetchGroupIDsOfClient($id)`

Fetches the IDs of all groups that contain the client with the given ID.

Param `id`

ID of the client whose groups shall be returned.

Return Value

An array of client-IDs.

Data Manipulation Methods

`addVendorOS(@$valRows)`

Adds one or more vendor-OS to the database.

Param `valRows`

An array-ref containing hash-refs with the data of the new vendor-OS(es).

Return Value

The IDs of the new vendor-OS(es), `undef` if the creation failed.

`removeVendorOS(@$vendorOSIDs)`

Removes one or more vendor-OS from the database.

Param `vendorOSIDs`

An array-ref containing the IDs of the vendor-OSes that shall be removed.

Return Value

`1` if the vendorOS(es) could be removed, `undef` if not.

`changeVendorOS(@$vendorOSIDs, @$valRows)`

Changes the data of one or more vendor-OS.

Param `vendorOSIDs`

An array-ref containing the IDs of the vendor-OSes that shall be changed.

Param `valRows`

An array-ref containing hash-refs with the new data for the vendor-OS(es).

Return Value

`1` if the vendorOS(es) could be changed, `undef` if not.

`addInstalledPlugin($vendorOSID, $pluginName)`

Adds a freshly installed plugin to a vendor-OS.

Param `vendorOSID`

The id of the vendor-OS the given plugin has been installed into

Param `pluginName`

The name of the plugin that has been installed

Return Value

The ID of the new reference entry, `undef` if the creation failed.

`removeInstalledPlugin($vendorOSID, $pluginName)`

Removes a uninstalled plugin for a vendor-OS.

Param `vendorOSID`

The id of the vendor-OS the given plugin has been uninstalled from

Param `pluginName`

The name of the plugin that has been uninstalled

Return Value

1 if it worked, `undef` if it didn't.

`addExport(@$valRows)`

Adds one or more export to the database.

Param `valRows`

An array-ref containing hash-refs with the data of the new export(s).

Return Value

The IDs of the new export(s), `undef` if the creation failed.

`removeExport(@$exportIDs)`

Removes one or more export from the database.

Param `exportIDs`

An array-ref containing the IDs of the exports that shall be removed.

Return Value

`1` if the export(s) could be removed, `undef` if not.

`changeExport(@$exportIDs, @$valRows)`

Changes the data of one or more export.

Param `vendorOSIDs`

An array-ref containing the IDs of the exports that shall be changed.

Param `valRows`

An array-ref containing hash-refs with the new data for the export(s).

Return Value

`1` if the export(s) could be changed, `undef` if not.

`incrementGlobalCounter($counterName)`

Increments the global counter of the given name and returns the *old* value.

Param `counterName`

The name of the global counter that shall be bumped.

Return Value

The value the global counter had before it was incremented.

`changeGlobalInfo($id, $value)`

Sets the global info element specified by the given ID to the given value.

Param `id`

The ID specifying the global info you'd like to change.

Param `value`

The new value for the global info element.

Return Value

The value the global counter had before it was incremented.

`addSystem(@$valRows)`

Adds one or more systems to the database.

Param `valRows`

An array-ref containing hash-refs with the data of the new system(s).

Return Value

The IDs of the new system(s), `undef` if the creation failed.

`removeSystem(@$systemIDs)`

Removes one or more systems from the database.

Param `systemIDs`

An array-ref containing the IDs of the systems that shall be removed.

Return Value

`1` if the system(s) could be removed, `undef` if not.

`changeSystem(@$systemIDs, @$valRows)`

Changes the data of one or more systems.

Param `systemIDs`

An array-ref containing the IDs of the systems that shall be changed.

Param `valRows`

An array-ref containing hash-refs with the new data for the system(s).

Return Value

`1` if the system(s) could be changed, `undef` if not.

`setClientIDsOfSystem($systemID, @$clientIDs)`

Specifies all clients that should offer the given system for booting.

Param `systemID`

The ID of the system whose clients you'd like to specify.

Param `clientIDs`

An array-ref containing the IDs of the clients that shall be connected to the system.

Return Value

`1` if the system/client references could be set, `undef` if not.

`addClientIDsToSystem($systemID, @$clientIDs)`

Add one or more clients to the set that should offer the given system for booting.

Param `systemID`

The ID of the system that you wish to add the clients to.

Param `clientIDs`

An array-ref containing the IDs of the new clients that shall be added to the system.

Return Value

`1` if the system/client references could be set, `undef` if not.

`removeClientIDsFromSystem($systemID, @$clientIDs)`

Removes the connection between the given clients and the given system.

Param `systemID`

The ID of the system you'd like to remove groups from.

Param `clientIDs`

An array-ref containing the IDs of the clients that shall be removed from the system.

Return Value

`1` if the system/client references could be set, `undef` if not.

`setGroupIDsOfSystem($systemID, @$groupIDs)`

Specifies all groups that should offer the given system for booting.

Param `systemID`

The ID of the system whose groups you'd like to specify.

Param `groupIDs`

An array-ref containing the IDs of the groups that shall be connected to the system.

Return Value

`1` if the system/group references could be set, `undef` if not.

`addGroupIDsToSystem($systemID, @$groupIDs)`

Add one or more groups to the set that should offer the given system for booting.

Param `systemID`

The ID of the system that you wish to add the groups to.

Param `groupIDs`

An array-ref containing the IDs of the new groups that shall be added to the system.

Return Value

`1` if the system/group references could be set, `undef` if not.

`removeGroupIDsFromSystem($systemID, @$groupIDs)`

Removes the connection between the given groups and the given system.

Param `systemID`

The ID of the system you'd like to remove groups from.

Param `groupIDs`

An array-ref containing the IDs of the groups that shall be removed from the system.

Return Value

`1` if the system/group references could be set, `undef` if not.

`addClient(@$valRows)`

Adds one or more clients to the database.

Param `valRows`

An array-ref containing hash-refs with the data of the new client(s).

Return Value

The IDs of the new client(s), `undef` if the creation failed.

`removeClient(@$clientIDs)`

Removes one or more clients from the database.

Param `clientIDs`

An array-ref containing the IDs of the clients that shall be removed.

Return Value

`1` if the client(s) could be removed, `undef` if not.

`changeClient(@$clientIDs, @$valRows)`

Changes the data of one or more clients.

Param `clientIDs`

An array-ref containing the IDs of the clients that shall be changed.

Param `valRows`

An array-ref containing hash-refs with the new data for the client(s).

Return Value

`1` if the client(s) could be changed, `undef` if not.

`setSystemIDsOfClient($clientID, @$systemIDs)`

Specifies all systems that should be offered for booting by the given client.

Param `clientID`

The ID of the client whose systems you'd like to specify.

Param `systemIDs`

An array-ref containing the IDs of the systems that shall be connected to the client.

Return Value

`1` if the client/system references could be set, `undef` if not.

`addSystemIDsToClient($clientID, @$systemIDs)`

Adds some systems to the set that should be offered for booting by the given client.

Param `clientID`

The ID of the client to which you'd like to add systems to.

Param `systemIDs`

An array-ref containing the IDs of the new systems that shall be added to the client.

Return Value

`1` if the client/system references could be set, `undef` if not.

`removeSystemIDsFromClient($clientID, @$systemIDs)`

Removes some systems from the set that should be offered for booting by the given client.

Param `clientID`

The ID of the client to which you'd like to remove systems from.

Param `systemIDs`

An array-ref containing the IDs of the systems that shall be removed from the client.

Return Value

`1` if the client/system references could be set, `undef` if not.

`setGroupIDsOfClient($clientID, @$groupIDs)`

Specifies all groups that the given client shall be part of.

Param `clientID`

The ID of the client whose groups you'd like to specify.

Param `groupIDs`

An array-ref containing the IDs of the groups that the client should be part of.

Return Value

`1` if the client/group references could be set, `undef` if not.

`addGroupIDsToClient($clientID, @$groupIDs)`

Adds the given client to the given groups.

Param `clientID`

The ID of the client that you'd like to add to the given groups.

Param `groupIDs`

An array-ref containing the IDs of the groups that shall be added to the client.

Return Value

`1` if the client/group references could be set, `undef` if not.

`removeGroupsIDsFromClient($clientID, @$groupIDs)`

Removes the given client from the given groups.

Param `clientID`

The ID of the client that you'd like to remove from the given groups.

Param `groupIDs`

An array-ref containing the IDs of the groups that shall be removed from the client.

Return Value

`1` if the client/group references could be set, `undef` if not.

`addGroup(@$valRows)`

Adds one or more groups to the database.

Param `valRows`

An array-ref containing hash-refs with the data of the new group(s).

Return Value

The IDs of the new group(s), `undef` if the creation failed.

`removeGroup(@$groupIDs)`

Removes one or more groups from the database.

Param `groupIDs`

An array-ref containing the IDs of the groups that shall be removed.

Return Value

`1` if the group(s) could be removed, `undef` if not.

`changeGroup(@$groupIDs, @$valRows)`

Changes the data of one or more groups.

Param `groupIDs`

An array-ref containing the IDs of the groups that shall be changed.

Param `valRows`

An array-ref containing hash-refs with the new data for the group(s).

Return Value

`1` if the group(s) could be changed, `undef` if not.

`setClientIDsOfGroup($groupID, @$clientIDs)`

Specifies all clients that should be part of the given group.

Param `groupID`

The ID of the group whose clients you'd like to specify.

Param `clientIDs`

An array-ref containing the IDs of the clients that shall be part of the group.

Return Value

`1` if the group/client references could be set, `undef` if not.

`addClientIDsToGroup($groupID, @$clientIDs)`

Add some clients to the given group.

Param `groupID`

The ID of the group to which you'd like to add clients.

Param `clientIDs`

An array-ref containing the IDs of the clients that shall be added.

Return Value

`1` if the group/client references could be set, `undef` if not.

`removeClientIDsFromGroup($groupID, @$clientIDs)`

Remove some clients from the given group.

Param `groupID`

The ID of the group from which you'd like to remove clients.

Param `clientIDs`

An array-ref containing the IDs of the clients that shall be removed.

Return Value

`1` if the group/client references could be set, `undef` if not.

`setSystemIDsOfGroup($groupID, @$systemIDs)`

Specifies all systems that should be offered for booting by the given group.

Param `groupID`

The ID of the group whose systems you'd like to specify.

Param `systemIDs`

An array-ref containing the IDs of the systems that shall be connected to the group.

Return Value

`1` if the group/system references could be set, `undef` if not.

`addSystemIDsToGroup($groupID, @$systemIDs)`

Adds some systems to the set that should be offered for booting by the given group.

Param `groupID`

The ID of the group to which you'd like to add systems.

Param `systemIDs`

An array-ref containing the IDs of the systems that shall be added.

Return Value

`1` if the group/system references could be set, `undef` if not.

`removeSystemIDsFromGroup($groupID, @$systemIDs)`

Removes some systems from the set that should be offered for booting by the given group.

Param `groupID`

The ID of the group from which you'd like to remove systems.

Param `systemIDs`

An array-ref containing the IDs of the systems that shall be removed.

Return Value

`1` if the group/system references could be set, `undef` if not.

`emptyDatabase()`

Removes all data from the database - the tables stay, but they will be empty.

Return Value

none

Data Aggregation Methods

`mergeDefaultAttributesIntoSystem($system)`

merges vendor-OS-specific plugin attributes and default system attributes into the given system hash, and pushes the default client attributes on top of that.

Param `system`

The system whose attributes shall be merged into (completed).

Return Value

none

`mergeDefaultAndGroupAttributesIntoClient($client)`

merges default and group configurations into the given client hash.

Param `client`

The client whose attributes shall be merged into (completed).

Return Value

none

`aggregatedSystemIDsOfClient($client)`

Returns an aggregated list of system-IDs that this client should offer for booting (as indicated by itself, the default client and the client's groups)

Param `client`

The client whose aggregated systems you're interested in.

Return Value

A list of unqiue system-IDs.

`aggregatedClientIDsOfSystem($system)`

Returns an aggregated list of client-IDs that offer this system for booting (as indicated by itself, the default system and the system's groups)

Param `system`

The system whose aggregated clients you're interested in.

Return Value

A list of unqiue client-IDs.

`aggregatedSystemFileInfoFor($system)`

Returns aggregated information about the kernel and initialramfs this system is using.

Param `system`

The system whose aggregated info you're interested in.

Return Value

A hash containing detailled info about the vendor-OS and export used by this system, as well as the specific kernel-file and export-URI being used.

Support Functions

`mergeAttributes($target, $source)`

Copies all attributes from source that are unset in target over (source extends target).

Param `target`

The hash to be used as copy target.

Param `source`

The hash to be used as copy source.

Return Value

none

`pushAttributes($target, $source)`

Copies all attributes that are set in source into the target (source overrules target).

Param `target`
The hash to be used as copy target.

Param `source`
The hash to be used as copy source.

Return Value
none

`externalIDForSystem($system)`

Returns the given system's name as an external ID - worked into a state that is usable as a filename.

Param `system`
The system you are interested in.

Return Value
The external ID (name) of the given system.

`externalIDForClient($client)`

Returns the given client's MAC as an external ID - worked into a state that is usable as a filename.

Param `client`
The client you are interested in.

Return Value
The external ID (MAC) of the given client.

`externalConfigNameForClient($client)`

Returns the given client's name as an external ID - worked into a state that is usable as a filename.

Param `client`
The client you are interested in.

Return Value
The external name of the given client.

`generatePlaceholdersFor($varName)`

Returns the given variable as a placeholder - surrounded by '@@@' markers.

Param `varName`
The variable you are interested in.

Return Value
The given variable as a placeholder string.