

## AXIGEN CliClient API v7.3.1

=====

### Content of this document:

1. Introduction
2. Software requirements
3. Files and directories
4. Installation
5. Check for PHP Threads support

### 1. Introduction

=====

**CLIClient is an API that allows easy Axigen mailserver configuration. CLI is an Axigen service that allows configuration of Axigen server. To access CLI, one must enable it and telnet on the port the service binds to, usually 7000. CLI uses contexts in order to group together similar operations, like server configuration (the "server#" context), pop3 service configuration (the "server-pop3#" context) or domain configuration (the "domain#" context). These contexts are positioned in a tree-like structure, so if one wants to configure something, one must go to the necessary context, make the changes through the use of commands and then commit the changes.**

**The CLIClient assigns objects to the various CLI contexts. These objects embed all the properties specific to a CLI context; for example the "domain" object has the "name" property representing the domain name, like "mydomain.com". Also, the CLIClient has various methods for getting and setting the object properties. Once changes on the object are finalized, the object configuration can be saved back to Axigen, thus committing the changes. Also, the CLIClient has methods that return other objects, for example from the "account" object one can obtain the "quotas" object, representing the restrictions for that specific account. Naturally, Axigen's CLI service must be enabled for the CLIClient library to configure that server.**

### 2. Software requirements

=====

**The software requirements, for Axigen Cliclient API, are:**

- \* Linux OS, kernel 2.4/2.6
- \* Windows 2003, Windows XP
- \* IIS 6.0 with fastCGI (you need to use php Non-thread-safe)
- \* Apache 2.2
- \* PHP 5.2.1 - 5.2.12, 5.3.0

### 3. Files and directories

=====

- \* cliclient.so/dll - needs to be located in php extension directory

- \* php files - need to be located in php include path

#### 4. Instalation

=====

- \* edit your php.ini
- \* search for extension\_dir and see where the path is set to
- \* search for include\_path
- \* append to include path ":/somepath/AxigenCliApi/"
- \* append the following line to your php.ini:  
extension = "cliclient.so"
- \* check php if is compile with threads or not to see what cliclient to copy
- \* check your OS, CPU to see what cliclient library to copy
- \* mv cliclient\_(OS)\_(CPU)\_(THREADS).so to the path from extension\_dir/cliclient.so
- \* tar zxvf phpfiles.tgz -C /somepath/AxigenCliApi/
- \* restart your apache server
- \* to start using AxigenCliApi include "Axigen.php" and check the Cli Object Documentation on how to use it.

#### 5. Check for PHP Threads support

=====

You can check if your php has Threads support by following these steps:

- \* php cli:
  - run this command in a shell:  
php --info | grep "Threads Safety"
- \* php apache:
  - create a php file containing <?php phpinfo(); ?>
  - fetch that file via apache and find the "Threads Safety" string
- If "Threads Safety" is enabled, use the library with threads, otherwise, use the one with no threads.



Object: Aacl Parents: Axigen->configAacl
--

**Method Name: updateAdminGroup**

Description: updates an admin group

Parameters:

string : GroupName ( the group name that you want to update )

**Method Name: updateAdminUser**

Description: updates an admin user

Parameters:

string : UsserName ( the user name that you want to update )

**Method Name: listResources**

Description: this method returns an array with all the available resources

Return: array

**Method Name: listAdminGroups**

Description: this method returns an array with all the available admin groups

Return: array

**Method Name: listAdminUsers**

Description: this method returns an array with all the available admin users

Return: array

**Method Name: listServerPrincipals**

Description: this method returns an array with all the available server principals

Return: array

**Method Name: listDomainPrincipals**

Description: this method returns an array with all the available domain principals (per specific domain)

Return: array

Parameters:

string : DomainName ( the domain name for which you want to see the principals )

**Method Name: purgeResources**

Description: this method purges all resources

Return: bool

**Method Name: addAdminGroup**

Description: adds a new admin group

Return: bool

Parameters:

string : GroupName ( the new group name )

**Method Name: removeAdminGroup**

Description: removes an admin group

Return: bool

Parameters:

string : GroupName ( the group name that will be removed )

: Forced ( Force group removal )

**Method Name: addAdminUser**

Description: adds a new admin user

Return: bool

Parameters:

string : UserName ( the new user name )

string : Password ( User password )

**Method Name: removeAdminUser**

Description: this method removes an admin user

Return: bool

Parameters:

string : UserName ( the user name to be removed )

```
Object: AccessControl
Parents:
  Cli->configAccessControl
  FtpBackup->configAccessControl
  Imap->configAccessControl
  ImapProxy->configAccessControl
  Pop3->configAccessControl
  Pop3Proxy->configAccessControl
  Smtpln->configAccessControl
  Webadmin->configAccessControl
  Webmail->configAccessControl
  WebmailProxy->configAccessControl
```

**Members:**

```
set/get maxConnections - maximum number of connections
set/get timeInterval - the time interval
set/get maxIntervalConnections - maximum number of connections in time interval
set/get peerMaxConnections - maximum number of connections from a single host
set/get peerTimeInterval - the time interval - single host
set/get peerMaxIntervalConnections - maximum number of connections in time interval - single host
```

**Method Name: updateAllowRule**

Description: this method returns an object with an allow rule loaded

Parameters:

```
string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )
```

**Method Name: listAllowRules**

Description: returns an array with all the allow rules

Return: array

**Method Name: listDenyRules**

Description: returns an array with all the deny rules

Return: array

**Method Name: addDenyRule**

Description: adds a deny rule

Return: bool

Parameters:

```
string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )
```

```
string (yes/no) : RuleEnable ( enable rule (optional default: yes) )
```

```
int : RulePriority ( rule priority (optional default: auto) )
```

**Method Name: removeDenyRule**

Description: removes a deny rule

Return: bool

Parameters:

```
string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )
```

**Method Name: updateDenyRule**

Description: this method updates a deny rule

Parameters:

```
string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )
```

```
string (yes/no) : Enable ( enable/disable rule )
```

```
int : Priority ( set the rule priority )
```

**Method Name: addAllowRule**

Description: adds an allow rule

Return: bool

Parameters:

```
string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )
```

```
string (yes/no) : RuleEnable ( enable rule (optional default: yes) )
```

```
int : RulePriority ( rule priority (optional default: auto) )
```

**Method Name: removeAllowRule**

Description: removes an allow rule

Return: bool

Parameters:

string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Account

Description: this is the object containing the Axigen account data for the selected account

Parents:

Domain->updateAccount

**Members:**

set/get name - the account name

set password - the account's password

set/get customerReference - the customer reference for use with external billing like systems

get accountClassName - the account class the selected user inherits settings from

get senderNotInAddressBookAction - this is the action selected by the user in Webmail to be executed if the email sender is not in his/her address book

**Method Name: configWebmailData**

Description: this method returns an object pertaining to WebmailData for the selected account

**Method Name: configWebmailAdvertising**

Description: this method returns an object pertaining to WebmailAdvertising for the selected account

**Method Name: configSendRecvRestrictions**

Description: This function returns an object of SendRecvRestrictions type for the selected account

**Method Name: configFilters**

Description: this method returns an object pertaining to Filters for the selected account

**Method Name: configContactInfo**

Description: this method returns an object pertaining to Contact Info for the selected account

**Method Name: configQuotas**

Description: this method returns an object pertaining to Quotas for the selected account

**Method Name: configLimits**

Description: this method returns an object pertaining to Limits for the selected account

**Method Name: listAliases**

Description: this method returns an array with all the aliases of the selected account

Return: array

**Method Name: addAlias**

Description: this method adds a new alias to the selected account

Return: bool

Parameters:

string : AliasName ( the new alias name )

**Method Name: removeAlias**

Description: this method removes an alias from the selected account

Return: bool

Parameters:

string : AliasName ( the name of the alias that you want to remove from the selected account )

**Method Name: registryInformation**

Description: this method returns an array with all the account registry information

Return: array

**Method Name: uploadWmFilter**

Description: this method uploads a Webmail Filter to the account

Return: bool

Parameters:

string : FilterData ( the filter data that you want to upload )

**Method Name: inheritSettingsFromDomainDefaults**

Description: this method sets the account settings' inheritance from the domain's defaults

Return: bool

**Method Name: inheritSettingsFromAccountClass**

Description: this method sets the account settings' inheritance from the account class

Return: bool

Parameters:

string : AccountClass ( the account class name that you want to set for the selected user )

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

Object: AccountClass Parents: Domain->updateAccountClass
--

**Members:**

set/get name - the account class name

---

**Method Name: configWebmailAdvertising**

Description: this method returns an object pertaining to WebmailAdvertising for the selected account class

---

**Method Name: configFilters**

Description: this method returns the Filters object for the selected account class

---

**Method Name: configQuotas**

Description: this method returns the Quotas object for the selected account class

---

**Method Name: configLimits**

Description: this method returns the Limits object for the selected account class

---

**Method Name: configSendRecvRestrictions**

Description: This function returns an object of SendRecvRestrictions type for the selected account class

---

**Method Name: save**

Description: this method saves the changed members of the account class

Return: bool

---

Object: ActiveFilters

Description: this is the object containing the ActiveFilters data for the selected filter

Parents:

Filters->updateActiveFilter

**Members:**

set/get priority - the priority of the filter

set/get filterName - the name of the filter as defined in the socket/script object set

set/get filterType - type of the filter (to which the object set belongs)

set/get applyOn - Specifies where this filter is activated (local, relay delivery or both) (available only on server filters)

set/get filteringOn - applies filter on levels: received|sent|sentAndReceived (available only on domain/accounts filters)

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: AdminGroup Parents: Aacl->updateAdminGroup
--

**Members:**

set/get name - gets group's name

set/get description - gets group's description

---

**Method Name: getEffectivePermissions**

Description: this method returns an object with all the effective permissions loaded

Parameters:

string : DomainName ( the domain name from where the permission will be loaded; if it is null, the server permissions will be loaded (this parameter is optional - server is default) )

---

**Method Name: getPermissions**

Description: this method returns an object with all the permissions loaded

Parameters:

string : DomainName ( the domain name from where the permission will be loaded; if it is null, the server permissions will be loaded (this parameter is optional - server is default) )

---

**Method Name: listParents**

Description: this method returns an array with all the group's parents

Return: array

---

**Method Name: listAvailableDomains**

Description: this method returns an array with all the available domains

Return: array

---

**Method Name: listMembers**

Description: this method returns an array with all the members

Return: array

---

**Method Name: addParentGroup**

Description: this method adds a new parent group

Return: bool

Parameters:

string : ParentGroupName ( the parent group that you want to add )

---

**Method Name: removeParentGroup**

Description: this method removes a parent group

Return: bool

Parameters:

string : ParentGroupName ( the parent group that you want to remove )

---

Object: AdminLimits  
Parents:  
Domain->configAdminLimits

**Members:**

set/get maxAccounts - maxAccounts on domain admin limits  
set/get maxMaillists - maxMaillists on domain admin limits  
set/get maxGroups - maxGroups on domain admin limits  
set/get maxFolderRecipients - maxFolderRecipients on domain admin limits  
set/get maxAccountClasses - maxAccountClasses on domain admin limits  
set/get maxAccountMessageSizeQuota - maxAccountMessageSizeQuota on domain admin limits  
set/get maxAccountMessageCountQuota - maxAccountMessageCountQuota on domain admin limits  
set/get maxAccountFolderCountQuota - maxAccountFolderCountQuota on domain admin limits  
set/get maxAccountFolderMessageSizeQuota - maxAccountFolderMessageSizeQuota on domain admin limits  
set/get maxAccountFolderMessageCountQuota - maxAccountFolderMessageCountQuota on domain admin limits  
set/get maxMaillistMessageSizeQuota - maxMaillistMessageSizeQuota on domain admin limits  
set/get maxMaillistMessageCountQuota - maxMaillistMessageCountQuota on domain admin limits  
set/get maxMaillistFolderCountQuota - maxMaillistFolderCountQuota on domain admin limits  
set/get maxMaillistFolderMessageSizeQuota - maxMaillistFolderMessageSizeQuota on domain admin limits  
set/get maxMaillistFolderMessageCountQuota - maxMaillistFolderMessageCountQuota on domain admin limits  
set/get maxPublicFolderMessageSizeQuota - maxPublicFolderMessageSizeQuota on domain admin limits  
set/get maxPublicFolderMessageCountQuota - maxPublicFolderMessageCountQuota on domain admin limits  
set/get maxPublicFolderFolderCountQuota - maxPublicFolderFolderCountQuota on domain admin limits  
set/get maxPublicFolderFolderMessageSizeQuota - maxPublicFolderFolderMessageSizeQuota on domain admin limits  
set/get maxPublicFolderFolderMessageCountQuota - maxPublicFolderFolderMessageCountQuota on domain admin limits  
set/get maxSubdomains - maxSubdomains on domain admin limits  
set/get subdomainStorageLocation - subdomainStorageLocation on domain admin limits  
set/get subdomainObjectStorageLocation - subdomainObjectStorageLocation on domain admin limits  
set/get enableSmtIncoming - the enable/disable status for SMTP Incoming  
set/get enableSmtOutgoing - the enable/disable status for SMTP Outgoing  
set/get enablePop3 - the enable/disable status for POP3  
set/get enableRpop - the enable/disable status for RPOP  
set/get enableImap - the enable/disable status for IMAP  
set/get enableWebmail - the enable/disable status for Webmail

**Method Name: save**

Description: this method must be called if one of the members has been changed  
Return: bool

**Method Name: listSubdomainMessageStorageLocations**

Description: list message storage locations for this sub-domain  
Return: array

**Method Name: addSubdomainMessageStorageLocation**

Description: adds a message storage location for this sub-domain  
Return: bool

Parameters:

string : Path ( the message storage location path )

**Method Name: removeSubdomainMessageStorageLocation**

Description: removes a message storage location for this sub-domain  
Return: bool

Parameters:

string : Path ( the message storage location path )

Object: AdminUser Parents: Aacl->updateAdminUser
--

**Members:**

set/get name - user's name  
set/get description - user's description  
set password - user's password

---

**Method Name: getEffectivePermissions**

Description: this method returns an object with all the effective permissions loaded

Parameters:

string : DomainName ( the domain name from where the permission will be loaded, if it's null the server permissions will be loaded (this parameter is optional - server is default) )

---

**Method Name: getPermissions**

Description: this method returns an object with all the permissions loaded

Parameters:

string : DomainName ( the domain name from where the permission will be loaded, if it's null the server permissions will be loaded (this parameter is optional - server is default) )

---

**Method Name: listParents**

Description: this method returns an array with all user's parents

Return: array

---

**Method Name: addParentGroup**

Description: this method adds a new parent group

Return: bool

Parameters:

string : ParentGroupName ( the parent group that you want to add )

---

**Method Name: removeParentGroup**

Description: this method removes a parent group

Return: bool

Parameters:

string : ParentGroupName ( the parent group that you want to remove )

---

**Method Name: listAvailableDomains**

Description: this method returns an array with all the available domains

Return: array

---

Object: AllowRule

Parents:

AccessControl->updateAllowRule

Listener->updateAllowRule

**Members:**

set/get ipSet - the ipSet parameter

set/get enable - the enable/disable status for the rule

set/get priority - the rule's priority

set/get maxConnections - maximum number of connections

set/get timeInterval - the time interval

set/get maxIntervalConnections - maximum number of connections in time interval

set/get peerMaxConnections - the set maximum number of connections from a single host

set/get peerTimeInterval - the time interval - single host

set/get peerMaxIntervalConnections - maximum number of connections in time interval - single host

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Axigen

Description: this object is the primary object for this wrapper; it establishes the connection to Axigen and from the resulting object you will be able to access all other objects

Note: this object must be initialized first

### Method Name: createDomain

Description: this method creates a new domain

Parameters:

string : DomainName ( the new domain name )

string : DomainLocation ( the location on disk )

string : PostmasterPassword ( the postmaster user password )

bool : Advanced ( if this is true, an object with the domain registration loaded will be returned, otherwise the command will be executed (default: false) )

Examples:

Example Create a new domain

```
<?php
```

```
require_once("Axigen.php");
```

```
$session = new Axigen();
```

```
/** Connecto to axigen server */
```

```
$session->connect("IP", "CLIPORT", "ADMINUSER", "ADMINPASSWORD");
```

```
/** Create a new domain on axigen server */
```

```
$result = $session->createDomain("localdomain", "domains/localdomain/", "postmasterPassword");
```

```
/** Check domain creation result */
```

```
if($result) {
```

```
    print "Domain has been created<br>";
```

```
/** Save configuration to axigen, for the domain to be registered on server restart */
```

```
$saveResult = $session->saveConfig();
```

```
/** Check save result */
```

```
if($saveResult) {
```

```
    print "Configuration saved<br>";
```

```
    } else {
```

```
        print "Unable to save configuration<br>";
```

```
    }
```

```
    } else {
```

```
        print "Unable to create domain<br>";
```

```
    }
```

```
?>
```

Result

this will print the domain creation result:

Domain has been created

Configuration saved

Example Create a new domain in advanced mode

```
<?php
```

```
require_once("Axigen.php");
```

```
$session = new Axigen();
```

```
/** Connecto to axigen server */
```

```
$session->connect("IP", "CLIPORT", "ADMINUSER", "ADMINPASSWORD");
```

```
/** Create a new domain on axigen server in advanced mode*/
```

```
$result = $session->createDomain("localdomain", "domains/localdomain/", "postmasterPassword", true);
```

```
/** set macl support */
```

```
$result->maclSupport = 1;
```

```
/** save domain */
```

```
$save = $result->save();

/** Check domain creation result */
if($save) {
    print "Domain has been created<br>";

    /** Save configuration to axigen, for the domain to be registered on server restart */
    $saveResult = $session->saveConfig();

    /** Check save result */
    if($saveResult) {
        print "Configuration saved<br>";
    } else {
        print "Unable to save configuration<br>";
    }
} else {
    print "Unable to create domain<br>";
}

?>
```

Result

this will print the domain creation result:

Domain has been created

Configuration saved

---

**Method Name: createSubDomain**

Description: this method creates a new sub-domain domain

Parameters:

string : Prefix ( sub-domain name (prefix for the domain) )

string : ParentDomain ( the parent domain for this sub-domain )

string : PostmasterPassword ( the postmaster user password )

bool : Advanced ( if this is true, an object with the domain registration loaded will be returned, otherwise the command will be executed (default: false) )

---

**Method Name: updateDomain**

Description: This function returns an object with the selected domain loaded

Parameters:

string : DomainName ( the domain name that you want to load )

---

**Method Name: editDomainName**

Description: This function returns an object with the selected domain loaded by name

Parameters:

string : DomainName ( the domain name that you want to load )

---

**Method Name: editDomainLocation**

Description: This function returns an object with the selected domain loaded by domain location

Parameters:

string : DomainLocation ( the domain location that you want to load )

---

**Method Name: configAacl**

Description: this function returns an object of AACL type

---

**Method Name: configServer**

Description: this function returns an object of Server type

---

**Method Name: viewQueue**

Description: this function returns an object of Queue type

---

**Method Name: disconnect**

Description: close connection with the server

Note: the connection with the server is closed on object destroy, so you don't need to explicit call disconnect

---

**Method Name: connect**

Description: this method is used to connect to the Axigen CLI

Note: please make sure that you have set the CLI listener to be able to connect to the place you run the script from

Return: bool

Parameters:

string : IpAddress ( IP address where the Axigen server is )

int : CliPort ( The CLI port from your Axigen configuration )

string : AdminUsername ( the admin user name )

string : Password ( the admin password )

bool : secure ( SSL connection (this parameter is optional, default: false) )

int : Timeout ( The connection time out (this parameter is optional) )

---

**Method Name: versions**

Description: returns an array with PHP files and the CliClient library version

Return: array

---

**Method Name: listDomains**

Description: this method returns an array with the enabled domains list filtered by parameter

Return: array

Parameters:

string : DomainFilter ( the domain filter (ex: domain\*: will list all domains that start with domain) (this parameter is optional) )

Examples:

Example List all the domains

```
<?php
```

```
require_once("Axigen.php");
```

```
$session = new Axigen();
```

```
/** Connecto to axigen server */
```

```
$session->connect("IP", "CLIPORT", "ADMINUSER", "ADMINPASSWORD");
```

```
/** get all the domains from axigen server*/
```

```
$domains = $session->listDomains();
```

```
/** print the resulted array */
```

```
var_dump($domains);
```

```
?>
```

Result

this will print all the axigen registered domains:

array

0 => 'localdomain1'

1 => 'localdomain2'

---

**Method Name: listAllDomains**

Description: this method returns an array with the all domains list filtered by parameter

Return: array

Parameters:

string : DomainFilter ( the domain filter (ex: domain\*: will list all domains that start with domain) (this parameter is optional) )

---

**Method Name: listDomainActivationRequests**

Description: this method returns an array with domain activation requests

Return: array

---

**Method Name: clearDomainActivationRequests**

Description: this method clears the domain activation request list

Return: bool

---

**Method Name: saveConfig**

Description: this method saves the Axigen Configuration on disk

Return: bool

---

**Method Name: axigenVersion**

Description: this method returns the Axigen version name

Return: string

Examples:

Example Connect to an axigen server and get server version

```
<?php
```

```
require_once("Axigen.php");
```

```
$session = new Axigen();
```

```
/** Connecto to axigen server */
```

```
$session->connect("IP", "CLIPORT", "ADMINUSER", "ADMINPASSWORD");
```

```
/** Get and print axigen version */
```

```
print $session->axigenVersion();
```

```
?>
```

Result

this will print something like:

6.1.1.23-win32

---

**Method Name: changePassword**

Description: this method changes the current user password

Return: bool

Parameters:

string : UserOldPassword ( Admin user old password )

string : UserNewPassword ( Admin user new password )

---

**Method Name: deleteDomainName**

Description: deletes a domain from the server using the domain name

Return: bool

Parameters:

string : DomainName ( The domain name that you want to delete )  
bool : dontKeepCopy ( delete the domain from you disk (optional, default: false) )

---

**Method Name: deleteDomainLocation**

Description: deletes a domain from the server using the domain location

Return: bool

Parameters:

string : DomainLocation ( The domain location that you want to delete )  
bool : dontKeepCopy ( delete the domain from you disk (optional, default: false) )

---

**Method Name: disableDomain**

Description: disables a domain from the server using the domain name

Return: bool

Parameters:

string : DomainName ( The domain name that you want to disable )

---

**Method Name: enableDomainName**

Description: enables a domain from the server using the domain name

Return: bool

Parameters:

string : DomainName ( The domain name that you want to enable )

---

**Method Name: enableDomainLocation**

Description: enables a domain from the server using the domain location

Return: bool

Parameters:

string : DomainLocation ( The domain location that you want to enable )

---

**Method Name: getLastError**

Description: this function returns an array with the last error

Return: array

---

Object: Cli  
Parents:  
Server->configCli

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the CLI service  
set/get minConnThreads - the minimum number of threads involved in I/O for the CLI service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get maxAuthCommands - the maximum number of commands the CLI service performs before the user logs in  
set/get logLevel - the log level for the CLI service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the CLI service

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the CLI configuration on the Axigen server

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: stop**

Description: This function stops the CLI service

Return: bool

**Method Name: start**

Description: This function starts the CLI service

Return: bool

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

Object: ContactInfo

Description: this is the object containing the contact info for the selected account

Parents:

Account->configContactInfo

---

**Members:**

set/get firstName - the first name for this user

set/get lastName - the last name for this user

set/get nickName - the nickname for this user

set/get personalEmail - the personal email address for this user

set/get businessEmail - the business email address for this user

set/get phone - the phone number for this user

set/get mobilePhone - the mobile phone number for this user

set/get homePhone - the home phone number for this user

set/get businessPhone - the business phone number for this user

set/get homeAddress - the home address for this user

set/get businessAddress - the business address for this user

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Debug

Object: Dnr  
Parents:  
Server->configDnr

**Members:**

set/get timeout - the default timeout nameservers have  
set/get retries - the default number of retries nameservers have  
set/get cacheSize - the cache size in KB for the DNR service  
set/get logLevel - the log level for the DNR service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the DNR service

**Method Name: save**

Description: This function saves the DNR configuration on the Axigen server

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Return: bool

**Method Name: stop**

Return: bool

**Method Name: addNameserver**

Description: This function adds a nameserver to the DNR list

Return: bool

Parameters:

String : address ( the address of the nameserver )  
int : priority ( the priority of the nameserver )  
int : timeout ( the number of seconds a request answer is awaited )  
int : retries ( the number of nameserver retries )

**Method Name: removeNameserver**

Description: This function removes a nameserver from the DNR list

Return: bool

Parameters:

int : priority ( the priority of the nameserver )

**Method Name: listNameservers**

Description: This function lists the nameservers from the DNR list

Return: Array

**Method Name: updateNameserver**

Description: This function updates a nameserver from the DNR list

Return: bool

Parameters:

int : priority ( the priority of the nameserver )  
String : newAddress ( the new value of the address )  
int : newPriority ( the new value of the priority )  
int : newTimeout ( the new value of the timeout )  
int : newRetries ( the new value of the number of retries )

Object: Domain

Description: this is the object containing the Axigen domain data for the selected domain

Parents:

Axigen->updateDomain

**Members:**

set/get name - the domain name  
set/get assignedIp - the domain assigned IP  
set/get appenderText - the domain appender text  
set/get enableSmtIncoming - SMTP Incoming status  
set/get enableSmtOutgoing - SMTP Outgoing status  
set/get enablePop3 - POP3 status  
set/get enableRpop - RPOP status  
set/get enableImap - IMAP status  
set/get enableWebmail - Webmail status  
set/get showWebmailLogin - displays this domain at Webmail login  
set/get publishRcptContacts - switch indicating if domain recipients contacts are published  
set/get enableAppender - the message appender  
set/get catchAllType - the type of Catch-All behavior  
set/get catchAllAccountName - the name of the account used to store all messages with unknown recipient  
set/get catchAllFolderName - the name of the folder used to store all messages with unknown recipient  
set/get enableLDAPSync - enables/disables LDAP-AXIGEN synchronization  
set/get ldapSyncConnectorName - the name of the LDAP connector to use  
set/get customerReference - the customer reference for use with external billing-like systems  
set/get defaultLanguage - the default language  
set/get defaultTimeZone - the default timezone  
set/get createUsersFromLdap - enabled/disabled auto-creation of users authenticated from LDAP  
set/get disposableMetadataQuotaThreshold - the disposable meta-data quota threshold  
get sharedFoldersNamespace - the shared folders namespace

**Method Name: updateAccount**

Description: this method returns an object with the selected account loaded

Parameters:

string : AccountName ( this is the account name that will be loaded )

**Method Name: updateGroup**

Description: this method returns an object with the selected group loaded

Parameters:

string : GroupName ( this is the group name that will be loaded )

**Method Name: updateFolderRcpt**

Description: this method returns an object with the selected Folder Rcpt loaded

Parameters:

string : FolderRcptName ( this is the Folder Rcpt name that will be loaded )

**Method Name: updateList**

Description: this method returns an object with the selected MailList loaded

Parameters:

string : ListName ( this is the list name that will be loaded )

**Method Name: updateAccountClass**

Description: this method returns an object with the selected Account Class loaded

Parameters:

string : AccountClassName ( this is the Account Class name that will be loaded )

**Method Name: configFilters**

Description: This function returns an object of Filters type for the selected domain

Note: these filters apply at domain level

**Method Name: configAdminLimits**

Description: This function returns an object of AdminLimits type for the selected domain

**Method Name: configAccountDefaultFilters**

Description: these filters are applied to the accounts that inherit filters from the domain Account Defaults Filters

Note: these filters are designed for those accounts that inherit filters from the domain Account Defaults Filters

**Method Name: configAccountDefaultLimits**

Description: these limits are applied to the accounts that inherit limits from the domain Account Defaults Limits

Note: these limits are designed for those accounts that inherit limits from the domain Account Defaults Limits

**Method Name: configAccountDefaultQuotas**

Description: these quotas are applied to the accounts that inherit quotas from the domain Account Defaults Quotas

Note: these quotas are designed for those accounts that inherit quotas from the domain Account Defaults Quotas

**Method Name: configAccountDefaultWebmailAdvertising**

Description: returns the account default Webmail advertising object

---

**Method Name: configPublicFolder**

Description: This function returns an object of PublicFolder type for the selected domain

---

**Method Name: configAccountDefaultSendRecvRestrictions**

Description: This function returns an object of SendRecvRestrictions type for the selected domain

---

**Method Name: configMigrationData**

Description: This function returns an object of MigrationData type for the selected domain

---

**Method Name: listAccounts**

Description: this method returns an array with the accounts list filtered by a certain parameter

Return: array

Parameters:

string : AccountFilter ( the account filter (ex: account\*: will list all the accounts that start with account) (this parameter is optional) )

Examples:

Example List all accounts from domain localdomain

```
<?php
require_once("Axigen.php");

$session = new Axigen();
/** Connecto to axigen server */
$session->connect("IP", "CLIPORT", "ADMINUSER", "ADMINPASSWORD");

/** Check if the domain exists */
$domains = $session->listDomains();
if(in_array("localdomain", $domains))
{
    $domainExists = true;
} else {
    print "The domain \"localdomain\" doesn't exist<br>";
    $domainExists = false;
}

/** If the domain exists, list all the accounts */
if($domainExists)
{
    /** load accounts list */
    $accounts = $session->updateDomain("localdomain")->listAccounts();

    /** print the resulted array */
    var_dump($accounts);
}
?>
```

Result

this will print the accounts from domain localdomain:

```
array
  0 => 'postmaster'
  1 => 'user1'
  2 => 'user2'
```

---

**Method Name: listAliases**

Description: this method returns an array with all the domain aliases

Return: array

---

**Method Name: listGroups**

Description: this method returns an array with the groups list filtered by a certain parameter

Return: array

Parameters:

string : GroupFilter ( the group filter (ex: group\*: will list all the groups that start with group) (this parameter is optional) )

---

**Method Name: listFolderRcpts**

Description: this method returns an array with a folderRcpts list filtered by a certain parameter

Return: array

Parameters:

string : FolderRcptFilter ( the folderRcpt filter (ex: folderRcpt\*: will list all the folderRcpts that start with folderRcpt) (this parameter is optional) )

---

**Method Name: listLists**

Description: this method returns an array with a groups list filtered by a certain parameter

Return: array

Parameters:

string : ListsFilter ( the Lists filter (ex: list\*: will list all the Lists that start with list) (this parameter is optional) )

**Method Name: listAccountClasses**

Description: this method returns an array with an Account Classes list filtered by a certain parameter

Return: array

Parameters:

string : AccountClassesFilter ( the AccountClasses filter (ex: accountclass\*: will list all the AccountClasses that start with accountclass) (this parameter is optional) )

---

**Method Name: registryInformation**

Description: this method returns an array with domain Registry Information

Return: array

---

**Method Name: storageInformation**

Description: this method returns an array with domain Storage Information

Return: array

---

**Method Name: addAccount**

Description: this method adds an account to the domain

Return: bool

Parameters:

string : AccountName ( this is the name for the new account )

string : AccountPassword ( this is the password for the new account )

---

**Method Name: removeAccount**

Description: this method removes an account from the domain

Return: bool

Parameters:

string : AccountName ( this is the account name that will be removed from the domain )

---

**Method Name: addGroup**

Description: this method adds a new Group to the domain

Return: bool

Parameters:

: GroupName ( )

---

**Method Name: removeGroup**

Description: this method removes a group from the domain

Return: bool

Parameters:

string : GroupName ( this is the group name that will be removed from the domain )

---

**Method Name: addFolderRcpt**

Description: this method adds a new FolderRcpt to the domain

Return: bool

Parameters:

string : FolderRcptName ( this is the FolderRcpt name that will be added to the domain )

---

**Method Name: removeFolderRcpt**

Description: this method removes a FolderRcpt from the domain

Return: bool

Parameters:

string : FolderRcptName ( this is the folderRcpt name that will be removed from the domain )

---

**Method Name: addList**

Description: this method adds a new List to the domain

Return: bool

Parameters:

string : ListName ( this is the list name that will be added to the domain )

string : ListPassword ( this is the list password )

string : ListAdminEmail ( this is the list admin email address )

---

**Method Name: removeList**

Description: this method removes a List from the domain

Return: bool

Parameters:

string : ListName ( this is the list name that will be removed from the domain )

---

**Method Name: addAccountClass**

Description: this method adds a new Account Class to the domain

Return: bool

Parameters:

string : AccountClassName ( this is the Account class name that will be added to the domain )

---

**Method Name: removeAccountClass**

Description: this method removes an Account Class from the domain

Return: bool

Parameters:

string : AccountClassName ( this is the Account Class name that will be removed from the domain )

---

**Method Name: addAlias**

Description: this method adds a new Alias to the domain

Return: bool

Parameters:

string : AliasName ( this is the Alias name that will be added to the domain )

---

**Method Name: removeAlias**

Description: this method removes an Alias from the domain

Return: bool

Parameters:

string : AliasName ( this is the Alias name that will be removed from the domain )

---

**Method Name: listStorages**

Description: lists all the storage registered on AXIGEN for this domain

Return: array

---

**Method Name: upgradeStorage**

Description: upgrades the storage representation of configuration data and folders to the latest version

Return: bool

Parameters:

bool : Forced ( forced upgrade (optional) )

---

**Method Name: repairAccounts**

Description: tries to recover lost accounts; the function will return the name of the newly-created account

Return: string

Parameters:

string : AccountName ( the account name that the server will try to repair )

---

**Method Name: syncContacts**

Description: synchronizes Domain Contacts folder with domain's objects contact information

Return: bool

---

**Method Name: migrateAccount**

Description: migrates an account from another server to a local account

Return: bool

Parameters:

string : LocalAccount ( the account name for migration )

string : RemoteHost ( the host of the migration server )

int : RemotePort ( the IMAP port of the migration server )

string : RemoteUser ( the IMAP username of the migration server )

string : RemoteUserPassword ( the IMAP password of the migration server )

string/array : IgnoreFolders ( specifies a list of folders which will not be migrated (either string for one folder, or an array for one or more) (optional parameter) )

bool : OverrideQuota ( specifies whether the mailbox quota should be overwritten (default: no) (optional parameter) )

bool : DeleteOriginal ( enables/disables the deletion of all migrated messages on the remote server (default: no) (optional parameter) )

bool : StructureOnly ( enables migration for the directory structure only (default: no) (optional parameter) )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: DomainRegistration  
Parents:  
Axigen->createDomain  
Axigen->createSubDomain  
Axigen->editDomainName  
Axigen->editDomainLocation

**Members:**

set/get name - domain name (available only on domain creation)  
set postmasterPassword - the password to the postmaster account (not available on domain registration)  
set/get domainLocation - the domain storage location  
set/get domainObjectsLocation - the domain objects (eg: accounts, mail lists etc.) storage location  
set/get maclSupport - enable/disable MACL support for this domain (deprecated)  
set/get groupwareSupport - enable/disable groupware support for this domain  
set/get prefix - sub-domain name (prefix for the domain) (available only on sub-domain creation)

**Method Name: save**

Description: this method must be called if one of the members has been changed  
Return: bool

**Method Name: listMessageStorageLocation**

Description: lists the domain messages storage location  
Return: array

**Method Name: addMessagesLocation**

Description: adds a messages storage location  
Return: bool  
Parameters:

string : Path ( the path to the messages storage )

**Method Name: removeMessagesLocation**

Description: removes a messages storage location  
Return: bool  
Parameters:  
string : Path ( the path to the messages storage )

---

Object: Filters

Description: this object contains the Filters data

Parents:

Account->configFilters

AccountClass->configFilters

FolderRcpt->configFilters

Group->configFilters

MailList->configFilters

Server->configFilters

Domain->configFilters

Domain->configAccountDefaultFilters

---

**Method Name: updateScriptFilter**

Description: this method returns an object of ScriptFilters type for the selected filter

Parameters:

string : ScripFilterName ( the name of the filter )

---

**Method Name: updateSocketFilter**

Description: this method returns an object of SocketFilters type for the selected filter

Parameters:

string : SocketFilterName ( the name of the filter )

---

**Method Name: updateActiveFilter**

Description: this method returns an object of ActiveFilters type for the selected filter

Parameters:

int : ActiveFilterPriority ( the priority of the filter )

---

**Method Name: listScriptFilters**

Description: returns an array with all the script filters

Return: array

---

**Method Name: addScriptFilter**

Description: this method adds a script filter

Return: bool

Parameters:

string : ScriptFilterName ( the name of the filter )

string : ScriptFilterFile ( the path to the file where the script is located )

---

**Method Name: removeScriptFilter**

Description: this method removes a script filter

Return: bool

Parameters:

string : ScriptFilterName ( the name of the filter )

---

**Method Name: listSocketFilters**

Description: returns an array with all the socket filters

Return: array

---

**Method Name: addSocketFilter**

Description: this method adds a socket filter

Return: bool

Parameters:

string : SocketFilterName ( the name of the filter )

string : SocketAddress ( the address of the filter, used to communicate with the filter )

string : SocketProtocolFile ( the path to the ASFL file that describes the communication protocol )

---

**Method Name: removeSocketFilter**

Description: this method removes a script filter

Return: bool

Parameters:

string : SocketFilterName ( the name of the filter )

---

**Method Name: listActiveFilters**

Description: returns an array with all the active filters

Return: array

---

**Method Name: addActiveFilter**

Description: this method adds an active filter

Return: bool

Parameters:

int : ActiveFilterPriority ( the priority of the filter )

string : ActiveFilterName ( the name of the filter as defined in the socket/script object sets )

string : ActiveFilterType ( type of the filter (to which object set belongs) (type: socket/script) )

---

**Method Name: removeActiveFilter**

Description: this method removes an active filter

Return: bool

Parameters:

int : ActiveFilterPriority ( the priority of the filter )

---

**Method Name: isInherited**

Description: checks if the filters are inherited

Return: int (1 if the value is inherited, 0 if not, or -1 if this does not apply to the current filters object)

---

**Method Name: inherit**

Description: set filters to be inherited from a selected source

Return: bool

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: FolderRcpt

Description: this is the object containing the Folder Rcpt data for the selected domain

Parents:

Domain->updateFolderRcpt

---

**Members:**

set/get name - the folder recipient's name

set/get mboxName - the mbox name set for this folder recipient

set/get enable - the enabled/disabled status for the folder recipient

---

**Method Name: configFilters**

Description: This function returns an object of Filters type for the selected folderrcpt

---

**Method Name: registryInformation**

Description: this method returns an array with folderrcpt Registry Information

Return: array

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: FtpBackup  
Parents:  
Server->configFtpBackup

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the Ftp Backup&Restore service  
set/get minConnThreads - the minimum number of threads involved in I/O for the Ftp Backup&Restore service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get logLevel - the log level for the FTP Backup&Restore service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the FTP Backup&Restore service  
set/get messagesTarEnabled - enabled/disabled tar archive on messages

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the Ftp Backup&Restore configuration on the Axigen server

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the FTP Backup&Restore service

Return: bool

**Method Name: stop**

Description: This function stops the FTP Backup&Restore service

Return: bool

**Method Name: getAccessControl**

Description: This function returns an access control object for this service

Return: AccessControl

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

**Method Name: fileSystemMount**

Description: mount a backup mount point

Return: bool

Parameters:

string : MountPoint ( mount point )

string : Domain ( domain to mount )

**Method Name: fileSystemMountObject**

Description: mount a backup mount point

Return: bool

Parameters:

string : MountPoint ( mount point )

string : Domain ( domain to mount )

string : Object ( Object to mount )

string : ObjectFolder ( ObjectFolder to mount (optional) )

**Method Name: fileSystemMountPublicFolder**

Description: mount a backup mount point

Return: bool

Parameters:

string : MountPoint ( mount point )

string : Domain ( domain to mount )

string : PublicFolder ( public folder to mount )

---

**Method Name: fileSystemUnmount**

Description: unmount a backup mount point

Return: bool

Parameters:

string : MountPoint ( mount point )

---

Object: Group

Description: this object contains the Group data for the selected group

Parents:

Domain->updateGroup

**Members:**

set/get name - the group's name - only usable in an UPDATE operation

set/get enable - the enabled/disabled status for the group

---

**Method Name: configFilters**

Description: this method returns an object of Filters type for the selected group

---

**Method Name: listAddresses**

Description: returns an array with addresses for the current group

Return: array

---

**Method Name: addAddress**

Description: adds an address for the group

Return: bool

Parameters:

string : Address ( The new address )

---

**Method Name: removeAddress**

Description: removes an address from the group

Return: bool

Parameters:

string : Address ( The address to be removed from group )

---

**Method Name: registryInformation**

Description: returns an array with registry information

Return: array

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Imap  
Parents:  
Server->configImap

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the IMAP service  
set/get minConnThreads - the minimum number of threads involved in I/O for the IMAP service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get allowStartTls - the secure connection (which start with STARTTLS)  
set/get secureConnectionAllowLogin - if plain text authentication is allowed on secure connections  
set/get secureConnectionAuthTypes - the allowed authentication methods on a secure connection  
set/get plainConnectionAuthTypes - the allowed authentication methods on a plain connection  
set/get plainConnectionAllowLogin - if plain text authentication is allowed on plain connections  
set/get userdbConnectorType - the userdb connector type  
set/get userdbConnectorName - the userdb connector name  
set/get logLevel - the log level for the IMAP service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the IMAP service  
set/get greetingsText - the greeting text for this service

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the IMAP configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the IMAP service

Return: bool

**Method Name: stop**

Description: This function stops the IMAP service

Return: bool

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

Object: ImapProxy  
Parents:  
Server->configImapProxy

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the IMAP Proxy service  
set/get minConnThreads - the minimum number of threads involved in I/O for the IMAP Proxy service  
set/get maxErrors - the maximum protocol errors number after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get allowStartTls - the secure connection (which start with STARTTLS)  
set/get authenticateOnProxy - if the authentication should be also done on proxy  
set/get mappingDataUsermapName - the mapping usermap name for the proxy  
set/get mappingDataMappingHost - the mapping data host for the proxy  
set/get mappingDataMappingPort - the mapping data port for the proxy  
set/get connectionDataTimeout - the timeout for the read/write operation  
set/get connectionDataMaxConnections - the maximum number of simultaneous connections  
set/get connectionDataSslEnable - if the connection should be SSL  
set/get connectionDataLocalInterface - the address of the interface used to make the connections  
set/get userdbConnectorType - the userdb connector type  
set/get userdbConnectorName - the userdb connector name  
set/get logLevel - the log level for the IMAP Proxy service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the IMAP Proxy service  
set/get greetingsText - the greeting text for this service

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the IMAP Proxy configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the IMAP service

Return: bool

**Method Name: stop**

Description: This function stops the IMAP service

Return: bool

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

Object: LdapConnector

Parents:

UserDb->updateLdapConnector

**Members:**

set/get name - the Ldap connector name.  
set/get hostUrl - the Ldap host URL (deprecated)  
set/get bindDN - the distinguished name for binding  
set/get bindPass - the password for binding  
set/get searchBase - the formatted string with the base where to search for the user (deprecated)  
set/get searchPattern - the formatted string with the user search pattern (deprecated)  
set/get passwordField - the name of the field containing the password (deprecated)  
set/get axigenHostField - the name of the field containing the Axigen server hostname  
set/get useFirst - the first returned field to be used if more are found (deprecated)  
set/get ldapURI1 - the first URI to use  
set/get ldapURI2 - the second URI to use  
set/get ldapURI3 - the third URI to use  
set/get ldapURI4 - the fourth URI to use  
set/get ldapURI5 - the fifth URI to use  
set/get synchronizationDirection - <choice> - values: axigenToLdap | ldapToAxigen | bothWays  
set/get synchronizationConflictResolution - <choice> - values: axigenWins | ldapWins | noChange  
set/get serverType - <choice> - values: OpenLdap | ActiveDirectory  
set/get timeout - the connection timeout  
set/get accountBaseDN - the account base DN  
set/get groupBaseDN - the group base DN  
set/get useCustomSchema - the enable/disable usage of a custom schema  
set/get customSchemaFile - <path> to the custom schema file  
set/get pollingInterval - the interval to check for changes in Ldap  
set/get transientErrorRetryInterval - retry interval for temporary errors  
set/get ignoreLDAPDeletes - enabled/disabled the LDAP deletes to be synchronized in axigen  
set/get clusteredSetup - enabled/disabled clustered setup

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Limits

Description: this object contains the Limits data

Parents:

Account->configLimits

AccountClass->configLimits

MailList->configLimits

Domain->configAccountDefaultLimits

### Members:

set/get sentMessagesCount - the maximum number of mails a user can send in a specified interval  
set/get sentMessagesSize - the maximum size of mails a user can send in a specified interval  
set/get sentMessagesInterval - the specified interval in seconds  
set/get pop3ConnectionCount - the number of simultaneous POP3 connections  
set/get imapConnectionCount - the number of simultaneous IMAP connections  
set/get webmailRcptCount - the maximum number of recipients for an email composed using Webmail  
set/get webmailSessionCount - the maximum simultaneous Webmail sessions  
set/get webmailAttSize - the size limit of an attachment that can be uploaded using Webmail  
set/get webmailAttCount - the maximum number of attachments for a mail composed in Webmail  
set/get webmailMessageSize - the size limit for a mail (body + attachments) composed in Webmail  
set/get rpopConnectionCount - the maximum number of RPOP connections a user is allowed to define  
set/get rpopRetrievalInterval - the minimum time interval (in minutes) between two RPOP retrievals on the same connection  
set/get maillistUserCount -  
set/get passwordEnforcementEnabled - the enabled/disabled status for the Password Enforcement policy  
set/get minimumPasswordLength - the minimum size of the password  
set/get maximumPasswordLength - the maximum size of the password  
set/get requiredPasswordCharacters - the required characters for setting a password  
set/get htmlFilterLevel - the security level for a HTML mail body (0-3)  
set/get temporaryAliasesCount - the temporary email address count limit  
set/get temporaryAliasesInterval - the temporary email address duration limit  
set/get enableProcessing - the enabled/disabled status for Processing  
set/get enableSmtIncoming - the enabled/disabled status for SMTP Incoming  
set/get enableSmtOutgoing - the enabled/disabled status for SMTP Outgoing  
set/get enablePop3 - the enabled/disabled status for POP3  
set/get enablePop3Proxy - the enabled/disabled status for POP3 Proxy  
set/get enableRpop - the enabled/disabled status for RPOP  
set/get enableImap - the enabled/disabled status for IMAP  
set/get enableImapProxy - the enabled/disabled status for IMAP Proxy  
set/get enableWebmail - the enabled/disabled status for Webmail  
set/get enableWebmailProxy - the enabled/disabled status for Webmail Proxy  
set/get enableCli - the enabled/disabled status for CLI  
set/get enableLog - the enabled/disabled status for Log  
set/get enableReporting - the enabled/disabled status for Reporting  
set/get enableFtpBackup - the enabled/disabled status for Ftp Backup  
set/get enableActiveSync - enable/disable ActiveSync?  
set/get activeSyncOptions - ActiveSync? options. Values: email contact pim (separated by space)  
set/get enableOutlookConnector - enabled/disabled Outlook Connector  
set/get enableWebPersonalOrganizer - enabled/disabled Web Personal Organizer  
set/get quotaSendMessageThreshold - the percent of used quota which blocks user from sending messages (available only for maillist)  
set/get passExpirationEnabled - enabled/disabled password expiration  
set/get passExpirationInterval - the interval of time in days that may pass from the moment the password was changed until the moment it expires and can not be used  
set/get passRenewalInterval - the interval of time that must pass from the moment of the last change of the password until a new change can be performed  
set/get passExpirationWarningEnabled - enabled/disabled sending of a warning message in case of password expiration  
set/get passExpirationWarningInterval - the interval of time in days after which warning message is sent  
set/get passExpirationWarningSubject - the subject template of the email message sent as warning for password expiration  
set/get passExpirationWarningSentDaily - enabled/disabled the password expiration warning to be sent every day during the expiration warning interval  
set/get passHistorySize - the number of consecutive changed passwords that can not be used as new password when a change of password is performed  
set/get passUnchangeable - enabled/disabled the capability for an user to change his/her password  
set/get welcomeMessageEnabled - enabled/disabled the post of a welcome message in an account's mailbox  
set/get welcomeMessageSubject - the subject template of the email message sent as welcome message  
set/get passExpirationWarningBody - the body template of the email message sent as warning for password expiration

set/get welcomeMessageBody - the body template of the email message sent as welcome message

set/get allowIdentityConfirmation - enabled/disabled the use of Identity Confirmation

---

**Method Name: isInherited**

Description: checks if the variable is inherited

Note: returns (1 if the value is inherited, 0 if not, or -1 if this does not apply to the current limits object)

Return: int

Parameters:

string : VariableName ( the variable )

---

**Method Name: inherit**

Description: the variable to inherit from selected source

Return: bool

Parameters:

string : VariableName ( the variable )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Listener  
Parents:  
Cli->updateListener  
FtpBackup->updateListener  
Imap->updateListener  
ImapProxy->updateListener  
Pop3->updateListener  
Pop3Proxy->updateListener  
Report->updateListener  
SmtIn->updateListener  
Webadmin->updateListener  
Webmail->updateListener  
WebmailProxy->updateListener

**Members:**

set/get address - the listener's address  
set/get enable - the enabled/disabled status of the listener  
set/get maxConnections - the maximum number of connections  
set/get timeInterval - the time interval  
set/get maxIntervalConnections - the maximum number of connections in a time interval  
set/get peerMaxConnections - the maximum number of connections from a single host  
set/get peerTimeInterval - the time interval - single host  
set/get peerMaxIntervalConnections - the maximum number of connections in a time interval - single host  
set/get idleTimeout - the inactivity timeout  
set/get sslEnable - SSL as enabled/disabled on the listener  
set/get listenerDomain - the domain of the listener

**Method Name: configSslControl**

Description: this method returns an object to configure SSL control for the listener

**Method Name: updateAllowRule**

Description: this method returns an object with allow rule loaded

Parameters:

string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )

**Method Name: listAllowRules**

Description: returns an array with all the allow rules

Return: array

**Method Name: listDenyRules**

Description: returns an array with all the deny rules

Return: array

**Method Name: addDenyRule**

Description: adds a deny rule

Return: bool

Parameters:

string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )

string (yes/no) : RuleEnable ( enables rule (optional default: yes) )

int : RulePriority ( rule priority (optional default: auto) )

**Method Name: removeDenyRule**

Description: removes a deny rule

Return: bool

Parameters:

string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )

**Method Name: updateDenyRule**

Description: this method updates a deny rule

Parameters:

string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )

string (yes/no) : Enable ( sets the enable/disabled status of the rule )

int : Priority ( sets the rule priority )

---

**Method Name: addAllowRule**

Description: adds an allow rule

Return: bool

Parameters:

string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )

string (yes/no) : RuleEnable ( enables rule (optional default: yes) )

int : RulePriority ( rule priority (optional default: auto) )

---

**Method Name: removeAllowRule**

Description: removes an allow rule

Return: bool

Parameters:

string : IpRangeNetwork ( IP range or network/mask (ex: 1.2.3.4-1.2.3.5 or 1.2.3.0/255.255.255.0) )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Log  
Parents:  
Server->configLog

**Members:**

set/get logLevel - the log level for the Log service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the Log service  
set/get path - the path for log service

**Method Name: updateListener**

Description: This function updates a listener from this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: save**

Description: This function saves the Log configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the Log service

Return: bool

**Method Name: stop**

Description: This function stops the Log service

Return: bool

**Method Name: listRules**

Description: This method lists the log rules

Return: array

**Method Name: addLogRule**

Description: This function adds a log rule to the log service

Return: bool

Parameters:

int : priority ( the priority of the log rule )

string : hostname ( the logging host address for this log rule. It will be used if the log type is remote )

string : module ( The 'value' parameter can be "\*" meaning every module or "pop3" for the pop3 module or "pop3 imap" for the POP3&IMAP modules )

int : level ( the log level for this rule )

string : fileName ( the file path for the logging file )

int : fileSize ( the maximum file size for the log file )

int : fileTime ( the maximum duration for the log file )

int : fileCount ( the maximum number of old log files that will be kept )

string : rotatePeriod ( the log rotate period. This will be override the log file file\_time. Accepted values for the 'value' parameters are: "none", "day", "week" or "month" )

string : type ( the type of the rule (choice: local|remote) )

**Method Name: updateRule**

Description: This function adds a log rule to the log service

Return: bool

Parameters:

int : RulePriority ( the priority of the log rule that you want to change )

int : priority ( the priority of the log rule )

string : hostname ( the logging host address for this log rule. It will be used if the log type is remote )

string : module ( The 'value' parameter can be "\*" meaning every module or "pop3" for the pop3 module or "pop3 imap" for the POP3&IMAP modules )

int : level ( the log level for this rule )

string : fileName ( the file path for the logging file )

int : fileSize ( the maximum file size for the log file )

int : fileTime ( the maximum duration for the log file )

int : fileCount ( the maximum number of old log files that will be kept )

string : rotatePeriod ( the log rotate period. This will be override the log file file\_time. Accepted values for the 'value' parameters are: "none", "day", "week" or "month" )

string : type ( the type of the rule (choice: local|remote) )

**Method Name: removeRule**

Description: removes a log rule

Return: bool

Parameters:

int : RulePriority ( the rule priority )

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:2001) )

---

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

---

Object: MailList

Description: this is the object containing the list data for the selected mail list

Parents:

Domain->updateList

**Members:**

set/get name - the name of the list  
set password - the list's mailbox access password  
set/get subscribeRcpt - the RCPT used for subscription  
set/get unsubscribeRcpt - the RCPT used for unsubscription  
set/get requestRcpt - the RCPT used for making a request  
set/get enabledRcpts - the RCPTs enabled for this list  
set/get description - the description of the list  
set/get adminConfirm - the adminConfirm parameter  
set/get senderAllow - the senderAllow parameter  
set/get moderate - the moderate parameter  
set/get ctypeAllow - the ctypeAllow parameter  
set/get adminEmail - the email for the admin  
set/get addHeader - the headers that will be added to the mail  
set/get bodyBegin - the text that will be inserted before the body  
set/get bodyEnd - the text that will be added to the body of the mail  
set/get unknownCommand - the description of the error in case of an unknown command  
set/get invalidUserName - the description of the error in case of an invalid user name  
set/get notAUser - the description of the error in case the user does not belong to the list  
set/get badConfirmation - the description of the error in case of a bad confirmation  
set/get userAlreadySubscribed - the description of the error in case the user already exists  
set/get invalidFormat - the description of the error in case of an invalid format  
set/get requestNeedsConfirmation - the template that will ask the user for a confirmation  
set/get requestNeedsAdminConfirmation - the template that will tell the user to wait for the admin's confirmation  
set/get autoRejectResponse - the template that will tell the user that his mail is rejected  
set/get welcome - the template that will tell the user that he has been created  
set/get goodbye - the template that will tell the user that he has been deleted  
set/get subscribeDenied - the template that will tell the user that he has not been created  
set/get enableSmtIncoming - the enabled/disabled status for SMTP Incoming  
set/get enableSmtOutgoing - the enabled/disabled status for SMTP Outgoing  
set/get enablePop3 - the enabled/disabled status for POP3  
set/get enableImap - the enabled/disabled status for IMAP  
set/get enableWebmail - the enabled/disabled status for Webmail  
set/get enableRpop - the enabled/disabled status for RPOP

**Method Name: configWebmailData**

Description: this method returns an object of WebmailData for the selected list

**Method Name: configFilters**

Description: this method returns an object of Filters for the selected list

**Method Name: configQuotas**

Description: this method returns an object of Quotas for the selected list

**Method Name: configLimits**

Description: this method returns an object of Limits for the selected list

**Method Name: listUsers**

Description: lists the available users for this list

Return: array

**Method Name: listRemoveHeaders**

Description: shows the list of headers that will be removed from a mail

Return: array

**Method Name: registryInformation**

Description: shows registry information

Return: array

**Method Name: addUser**

Description: adds a user to the list

Return: bool

Parameters:

string : userEmail ( New user email address )

string : userName ( New user name )

**Method Name: removeUser**

Description: removes an user from the list

Return: bool

Parameters:

string : UserEmail ( User email address to be removed )

---

**Method Name: updateUser**

Description: updates a user name

Return: bool

Parameters:

string : UserEmail ( User email address to be updated )

string : NewName ( )

---

**Method Name: addRemoveHeader**

Description: adds a header to the list of headers to be removed

Return: bool

Parameters:

string : Header ( Header to be added to the list )

---

**Method Name: removeRemoveHeader**

Description: removes a header from the list of headers to be removed

Return: bool

Parameters:

string : Header ( Header to be removed from the list )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Message

Object: MigrationData

Description: this is the object containing the Migration Data for the selected domain

Parents:

Domain->configMigrationData

---

**Members:**

set/get enable - the enabled status for the migration of accounts from this domain

set/get remoteImapHost - the name of the remote IMAP machine from which the domain's accounts are migrated

set/get remoteImapPort - the IMAP server's port on the remote machine

set/get remoteSmtphost - the name of the remote SMTP machine from which the domain's accounts are migrated

set/get remoteSmtport - the SMTP server's port on the remote machine

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Mobility Parents: Server->configMobility
--

**Members:**

set/get enableMobileWebmail - enable/disable Mobile Webmail

set/get enableActiveSync - enable/disable ActiveSync

---

**Method Name: save**

Description: This function saves the Mobility configuration back to Axigen

Return: bool

---

**Method Name: removeActiveSyncAccount**

Description: this method removes an ActiveSync Account from the server

Return: bool

Parameters:

string : AccountName ( this is the account name that will have its ActiveSync removed )

---

**Method Name: listActiveSyncAccounts**

Description: this method returns an array with ActiveSync Accounts list filtered by a certain parameter

Return: array

Parameters:

string : ActiveSyncAccountFilter ( the ActiveSync account filter (ex: account\*: will list all the accounts that start with account) (this parameter is optional)

)

---

Object: Permissions

Parents:

AdminGroup->getEffectivePermissions

AdminGroup->getPermissions

AdminUser->getEffectivePermissions

AdminUser->getPermissions

---

**Method Name: save**

Description: saves permissions

---

**Method Name: denyPermission**

Description: denies permission for the selected user or group

Note: is not available for effective permissions

Parameters:

string : PermissionName ( the permission name )

---

**Method Name: revokePermission**

Description: revokes permission for the selected user or group

Note: is not available for effective permissions

Parameters:

string : PermissionName ( the permission name )

---

**Method Name: grantPermission**

Description: grants a permission for the selected user or group

Note: is not available for effective permissions

Parameters:

string : PermissionName ( the permission name )

---

**Method Name: showPermissions**

Description: shows permissions for the selected user/group

Return: array

---

Object: Pop3  
Parents:  
Server->configPop3

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the POP3 service  
set/get minConnThreads - the minimum number of threads involved in I/O for the POP3 service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get allowStartTls - the secure connection (which starts with STARTTLS)  
set/get secureConnectionAllowLogin - if plain text authentication is allowed on secure connections  
set/get secureConnectionAuthTypes - the allowed authentication methods on a secure connection  
set/get plainConnectionAuthTypes - the allowed authentication methods on a plain connection  
set/get plainConnectionAllowLogin - if plain text authentication is allowed on plain connections  
set/get userdbConnectorType - the userdb connector type  
set/get userdbConnectorName - the userdb connector name  
set/get logLevel - the log level for the POP3 service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the POP3 service  
set/get greetingsText - the greeting text for this service

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the POP3 configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the POP3 service

Return: bool

**Method Name: stop**

Description: This function stops the POP3 service

Return: bool

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

Object: Pop3Proxy  
Parents:  
Server->configPop3Proxy

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the POP3 service  
set/get minConnThreads - the minimum number of threads involved in I/O for the POP3 service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get allowStartTls - the secure connection (which starts with STARTTLS)  
set/get authenticateOnProxy - if the authentication should be also done on proxy  
set/get mappingDataUsermapName - the mapping usermap name for the proxy  
set/get mappingDataMappingHost - the mapping data host for the proxy  
set/get mappingDataMappingPort - the mapping data port for the proxy  
set/get connectionDataTimeout - the timeout for the read/write operation  
set/get connectionDataMaxConnections - the maximum number of simultaneous connections  
set/get connectionDataSslEnable - if the connection should be SSL  
set/get connectionDataLocalInterface - the address of the interface used to make the connections  
set/get userdbConnectorType - the userdb connector type  
set/get userdbConnectorName - the userdb connector name  
set/get logLevel - the log level for the POP3 Proxy service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the POP3 Proxy service  
set/get greetingsText - the greeting text for this service

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the POP3 Proxy configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the POP3 service

Return: bool

**Method Name: stop**

Description: This function stops the POP3 service

Return: bool

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

Object: Processing Parents: Server->configProcessing
--

**Members:**

set/get maxMailScheduleInterval - the maximum interval for email rescheduling in case of a delivery error (in seconds)  
set/get queueScheduleInterval - the interval for queue checking  
set/get maxRetryCount - the maximum number of retries to deliver an email  
set/get deliveryThreads - the number of email delivery threads  
set/get internalQueueSize - the internal processing queue size (in number of messages)  
set/get messagesPerSecond - the maximum number of emails that a mailbox (folder) is allowed to receive in one second  
set/get mailboxDisableInterval - the number of seconds a mailbox (folder) is disabled if the "messages per second" limit is disabled  
set/get maxDirectoryCount - the maximum number of directories in the queue directory  
set/get queuePath - the path of the queue directory  
set/get logLevel - the log level for the processing service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the processing service  
set/get filteringThreads - number of threads handling the message filtering

---

**Method Name: save**

Description: This function saves the processing configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

---

Object: PublicFolder Description: this object contains the Public Folder data Parents: Domain->configPublicFolder
--

**Members:**

set/get name - the name of the public folder

---

**Method Name: configQuotas**

Description: this method returns quotas object

---

**Method Name: listMboxes**

Description: this method returns an array with all the mboxes

Return: array

---

**Method Name: addMbox**

Description: this method adds a new Mbox

Return: bool

Parameters:

string : MboxName ( The new mbox name )

---

**Method Name: removeMbox**

Description: this method removes a mbox

Return: bool

Parameters:

string : MboxName ( The mbox name to be removed )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Queue  
Parents:  
Axigen->viewQueue

**Method Name: listQueueEMails**

Description: returns an array with all the emails from the queue

Return: array

**Method Name: clearFilters**

Description: clears the already set filters

Return: bool

**Method Name: imposeSizeRestriction**

Description: sets the filter size restriction

Return: bool

Parameters:

int : MessageSize ( the message size )  
string : CompareType ( type of comparison (equal , less, more) )  
string : MessageSizeType ( type of size (B - bytes, KB - kilobytes, MB - megabytes) )

**Method Name: imposeStatusRestriction**

Description: sets the filter status restriction

Return: bool

Parameters:

string : EmailStatus ( the email status ("SEND FAILURE", "PROCESSING ERROR", "RELAY ERROR", "LOCAL ERROR", "FILTER REJECT", "FILTER DISCARD") )

**Method Name: imposeAgeRestriction**

Description: sets the filter age restriction

Return: bool

Parameters:

int : Days ( number of days )  
int : Hours ( number of hours )  
int : Minutes ( number of minutes )  
int : Seconds ( number of seconds )  
string : CompareType ( type of comparison (equal , less, more) )

**Method Name: imposeRetriesRestriction**

Description: sets the filter retries restriction

Return: bool

Parameters:

int : NrOfRetries ( number of retries )  
string : CompareType ( type of comparison (equal , less, more) )

**Method Name: imposeNextScheduleRestriction**

Description: sets the filter next schedule restriction

Return: bool

Parameters:

string : Date ( next schedule date (ex: 2 Jan 2007 17:34) )  
string : CompareType ( type of comparison (equal , less, more) )

**Method Name: imposeSenderRestriction**

Description: sets the filter sender restriction

Return: bool

Parameters:

string : Sender ( the Sender to filter )  
string : CompareType ( type of comparison (equal , not equal) )

**Method Name: imposeRecipientRestriction**

Description: sets the filter recipient restriction

Return: bool

Parameters:

string : Recipient ( the Recipient to filter )  
string : CompareType ( type of comparison (equal , not equal) )

**Method Name: deleteFilteredEmails**

Description: applies a delete action to filtered emails

Return: bool

**Method Name: sendNDRFilteredEmails**

Description: applies a NDR action to filtered emails

Return: bool

**Method Name: retryFilteredEmails**

Description: applies a retry action to filtered emails

Return: bool

---

**Method Name: listFilteredEMails**

Description: returns an array with emails filtered by already set filters

Return: array

---

**Method Name: forceQueueDelivery**

Description: this method forces the emails queue delivery

Return: bool

---

**Method Name: retryQueueEMail**

Description: this method tells Axigen to retry sending the mailId

Return: bool

Parameters:

int : MailID ( the mail it to retry )

---

**Method Name: sendNDRQueueEMail**

Description: this method tells Axigen to send NDR to the mailId

Return: bool

Parameters:

int : MailID ( the mail it to send NDR )

---

**Method Name: deleteQueueEMail**

Description: this method tells Axigen to delete the mailId

Return: bool

Parameters:

int : MailID ( the mailId to delete )

---

Object: Quotas

Description: this object contains the Quotas data

Parents:

Account->configQuotas

AccountClass->configQuotas

MailList->configQuotas

PublicFolder->configQuotas

Domain->configAccountDefaultQuotas

### Members:

set/get mboxCount - the maximum number of folders

set/get totalMessageCount - the maximum number of messages in all folders

set/get totalMessageSize - the maximum size (in KB) of all messages in all folders

set/get messageCount - the default maximum number of messages in a folder

set/get messageSize - the default maximum size (in KB) of messages in a folder

set/get limitThreshold - percent of quota which triggers an alert

set/get limitNotificationInterval - the minimum interval (in minutes) between two quota limit notifications

set/get limitNotificationEnabled - enabled/disabled the quota limit notification

set/get limitNotificationSubject - email subject of quota limit notification message

set/get limitNotificationBody - email body of quota limit notification message

set/get quotaSendMessageThreshold - percent of quota which blocks user from message sending

---

### Method Name: isInherited

Description: check if the variable is inherited

Return: int (1 if the value is inherited, 0 if not, or -1 if this does not apply to the current quota object)

Parameters:

string : VariableName ( the variable )

---

### Method Name: inherit

Description: sets the variable to inherit from the selected source

Return: bool

Parameters:

string : VariableName ( the variable )

---

### Method Name: save

Description: this method must be called if one of the members has been changed

Return: bool

---

### Method Name: listMboxes

Description: this method returns an array with all the mboxes and their quotas

Return: array

---

### Method Name: setMboxQuota

Description: this method sets quota on a mbox

Return: bool

Parameters:

string : MboxName ( the mbox name where you want to change quotas )

double : QuotaMessageCount ( the new message count quota )

double : QuotaMessageSize ( the new message size quota )

---

### Method Name: showMboxQuota

Description: this method returns an array with selected mbox quota

Return: array

Parameters:

string : MboxName ( the mbox name from where you want to retrieve quotas )

Object: Report  
Parents:  
Server->configReport

**Members:**

set/get DomainEnable - Enabled/Disabled Domain Collecting  
set/get DomainObjectEnable - Enabled/Disabled Domain Objects Collecting  
set/get reportingInterval - Collect samples every (values: 0-5844 days)  
set/get logLevel - the log level for the Reporting service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the Reporting service  
set/get SNMPEnable - the SNMP service as enabled  
set/get SNMPSentTrapsToAllManagers -  
set/get SNMPCommunity -

**Method Name: updateListener**

Description: This function updates a listener from this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: save**

Description: This function saves the Reporting configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the Reporting service

Return: bool

**Method Name: stop**

Description: This function stops the Reporting service

Return: bool

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

**Method Name: listSNMPTrapDestinations**

Description: lists available SNMP trap destinations

Return: array

**Method Name: addSNMPTrapDestination**

Description: adds a SNMP trap destination to the service

Return: bool

Parameters:

string : Address ( SNMP Trap Destination Address )

**Method Name: removeSNMPTrapDestination**

Description: removes a SNMP trap destination from the service

Return: bool

Parameters:

string : Address ( SNMP Trap Destination Address )

Object: Rpop Parents: Server->configRpop
--

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the RPOP service

set/get minConnThreads - the minimum number of threads involved in I/O for the RPOP service

set/get logLevel - the log level for the RPOP service

set/get logHost - the address of the host that offers log services

set/get logType - the log type for the RPOP service

---

**Method Name: save**

Description: This function saves the RPOP configuration back to Axigen

Return: bool

---

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

---

**Method Name: start**

Description: This function starts the RPOP service

Return: bool

---

**Method Name: stop**

Description: This function stops the RPOP service

Return: bool

---

Object: ScriptFilters

Description: this is the object containing the scriptfilters data for the selected filter

Parents:

Filters->updateScriptFilter

---

**Members:**

set/get name - gets the name of the filter

set/get type - gets the type of the script filter

set/get file - gets the path to the file where the script is located

---

**Method Name: uploadFilter**

Description: uploads a SIEVE filter to the Axigen server with the name pointed by the selected scriptfilter file name

Return: bool

Parameters:

string : FilterData ( the filter data that you want to upload )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: SendRecvRestrictions

Parents:

Account->configSendRecvRestrictions

AccountClass->configSendRecvRestrictions

Domain->configAccountDefaultSendRecvRestrictions

---

**Members:**

set/get enableSendRestrictions - enable/disable sending restrictions

set/get sendPolicy - values: 1 - accept | 0 - reject

set/get enableRecvRestrictions - enable/disable receiving restrictions

set/get recvPolicy - values: 1 - accept | 0 - reject

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

**Method Name: listSendPolicyExceptions**

Description: list send policy exceptions

Return: array

---

**Method Name: listRecvPolicyExceptions**

Description: list receiving policy exceptions

Return: array

---

**Method Name: addSendPolicyException**

Description: adds send policy exceptions

Return: bool

Parameters:

string : String ( the send policy exception string )

---

**Method Name: addRecvPolicyException**

Description: adds receive policy exceptions

Return: bool

Parameters:

string : String ( the receive policy exception string )

---

**Method Name: removeSendPolicyException**

Description: removes send policy exceptions

Return: bool

Parameters:

string : String ( the send policy exception string )

---

**Method Name: removeRecvPolicyException**

Description: removes receive policy exceptions

Return: bool

Parameters:

string : String ( the receive policy exception string )

---

Object: Server  
Parents:  
Axigen->configServer

**Members:**

set/get primaryDomain - the name of the Axigen primary domain  
set/get sslRandomFile - the path to the SSL random file  
set/get defaultLanguage - the default language  
set/get defaultTimeZone - the default timezone  
set/get serverName - the server name  
set/get enableIOSynchronization - the enabled/disabled status for I/O Synchronization  
set/get enableProcessing - the enabled/disabled status for Processing  
set/get enableSmtIncoming - the enabled/disabled status for SMTP Incoming  
set/get enableSmtOutgoing - the enabled/disabled status for SMTP Outgoing  
set/get enablePop3 - the enabled/disabled status for POP3  
set/get enablePop3Proxy - the enabled/disabled status for POP3 Proxy  
set/get enableRpop - the enabled/disabled status for RPOP  
set/get enableImap - the enabled/disabled status for IMAP  
set/get enableImapProxy - the enabled/disabled status for IMAP Proxy  
set/get enableWebmail - the enabled/disabled status for Webmail  
set/get enableWebmailProxy - the enabled/disabled status for Webmail Proxy  
set/get enableCli - the enabled/disabled status for CLI  
set/get enableLog - the enabled/disabled status for Log  
set/get enableReporting - the enabled/disabled status for Reporting  
set/get enableFtpBackup - the enabled/disabled status for Ftp Backup

**Method Name: configFilters**

Description: this method returns an object of Filters for the selected server

**Method Name: configLog**

Description: this method returns an object of Log for the selected server

**Method Name: configCli**

Description: this method returns an object of CLI for the selected server

**Method Name: configRpop**

Description: this method returns an object of RPOP for the selected server

**Method Name: configSmtIn**

Description: this method returns an object of SmtIn for the selected server

**Method Name: configSmtOut**

Description: this method returns an object of SmtOut for the selected server

**Method Name: configProcessing**

Description: this method returns an object of Processing for the selected server

**Method Name: configPop3**

Description: this method returns an object of POP3 for the selected server

**Method Name: configImap**

Description: this method returns an object of IMAP for the selected server

**Method Name: configWebmail**

Description: this method returns an object of Webmail for the selected server

**Method Name: configWebadmin**

Description: this method returns an object of Webadmin for the selected server

**Method Name: configFtpBackup**

Description: this method returns an object of FtpBackup for the selected server

**Method Name: configDnr**

Description: this method returns an object of DNR for the selected server

**Method Name: configReport**

Description: this method returns an object of Report for the selected server

**Method Name: configPop3Proxy**

Description: this method returns an object of Pop3Proxy for the selected server

**Method Name: configImapProxy**

Description: this method returns an object of ImapProxy for the selected server

**Method Name: configUserDb**

Description: this method returns an object of UserDb for the selected server

**Method Name: configWebmailProxy**

Description: this method returns an object of WebmailProxy for the selected server

**Method Name: configMobility**

Description: this method returns an object of Mobility for the selected server

**Method Name: updateUsermap**

Description: This method returns an object of usermap type

Parameters:

: name ( )

---

**Method Name: services**

Description: returns an array with all the services and their status

Return: array

---

**Method Name: startService**

Description: starts an Axigen service

Return: bool

Parameters:

string : ServiceName ( the name of the service that you want to start )

---

**Method Name: stopService**

Description: stops an Axigen service

Return: bool

Parameters:

string : ServiceName ( the name of the service that you want to stop )

---

**Method Name: addUsermap**

Description: This method is used at adding a usermap

Return: bool

Parameters:

: \$name ( )

: \$type ( )

: \$userdbConnectorType ( )

: \$userdbConnectorName ( )

: \$localFile ( )

---

**Method Name: listUsermaps**

Description: this method returns an array with the usermaps list

Return: array

---

**Method Name: removeUsermap**

Description: This method removes a usermap by name

Return: bool

Parameters:

: name ( )

---

**Method Name: save**Description: This function saves a server object back to Axigen

---

Object: Smtpln Parents: Server->configSmtpln
--

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the SMTP Incoming service  
set/get minConnThreads - the minimum number of threads involved in I/O for the SMTP Incoming service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get maxReceivedHeaders - the maximum number of headers that an email received by Axigen can have  
set/get enableSmtplRouting - the enabled/disabled status for SMTP Routing  
set/get mappingDataUsermapName - the mapping usermap name for the proxy  
set/get mappingDataMappingHost - the mapping data host for the proxy  
set/get mappingDataMappingPort - the mapping data port for the proxy  
set/get userdbConnectorType - the type of the userdb connector  
set/get userdbConnectorName - the name of the userdb connector  
set/get logLevel - the log level for the SMTP Incoming service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the SMTP Incoming service

---

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

---

**Method Name: save**

Description: This function saves the SMTP Incoming configuration back to Axigen

Return: bool

---

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

---

**Method Name: start**

Description: This function starts the SMTP Incoming service

Return: bool

---

**Method Name: stop**

Description: This function stops the SMTP Incoming service

Return: bool

---

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

---

Object: SmtOut Parents: Server->configSmtOut
--

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the SMTP Outgoing service

set/get minConnThreads - the minimum number of threads involved in I/O for the SMTP Outgoing service

set/get logLevel - the log level for the SMTP Outgoing service

set/get logHost - the address of the host that offers log services

set/get logType - the log type for the SMTP Outgoing service

---

**Method Name: save**

Description: This function saves the SMTP Outgoing configuration back to Axigen

Return: bool

---

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

---

**Method Name: start**

Description: This function starts the SMTP Outgoing service

Return: bool

---

**Method Name: stop**

Description: This function stops the SMTP Outgoing service

Return: bool

---

Object: SocketFilters

Description: this is the object containing the socketfilter data for the selected filter

Parents:

Filters->updateSocketFilter

**Members:**

set/get name - gets the name of the filter

set/get address - gets the address of the filter, used to communicate with the filter

set/get protocolFile - gets the path to the ASFL file that describes the communication protocol

set/get actionOnMatch - gets the action to be taken in case the filter matches an email

set/get maxConnections - gets the maximum number of connections that will be made to the filter

set/get idleTimeout - gets the inactivity timeout of the connection (in seconds)

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: SslControl  
Parents:  
Listener->configSslControl

**Members:**

set/get allowedVersions - the allowed SSL versions  
set/get maxChainDepth - the maximum depth of verification  
set/get cipherSuite - the cipher suite to be used  
set/get useEphemeralKey - use/not use ephemeral keys  
set/get certFile - path for certification chain file  
set/get caFile - path for Certificate Authority file  
set/get dhParamFile - path to Diffie-Hellman parameter file  
set/get requestClientAuth - request/not request client authentication

**Method Name: save**

Description: saves members? changes

Return: bool

---

Object: UdpListener Parents: Log->updateListener
--

**Members:**

set/get address - the address of the UDP listener

set/get enable - the enabled/disabled status of the UDP listener

**Method Name: save**

Description: This function saves the UdpListener configuration back to Axigen

Return: bool

---

Object: UserDb Parents: Server->configUserDb
--

**Members:**

set/get maxThreads - the maximum number of threads

set/get logLevel - the log level

set/get logHost - the address of the host that offers log services

set/get logType - the log type

---

**Method Name: updateLdapConnector**

Description: updates a Ldap connector

Parameters:

string : LdapConnectorName ( the name of the Ldap connector )

---

**Method Name: listLdapConnectors**

Description: lists the Ldap connectors

Return: array

---

**Method Name: addLdapConnector**

Description: adds a new Ldap connector

Return: bool

Parameters:

string : LdapConnectorName ( the name of the Ldap connector )

string : HostUrl ( the Ldap Host URL )

---

**Method Name: removeLdapConnector**

Description: removes a Ldap connector

Return: bool

Parameters:

string : LdapConnectorName ( the name of the Ldap connector )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Usermap Parents: Server->updateUsermap
--

**Members:**

set/get name - the usermap name

set/get type - the usermap type

set/get userdbConnectorName - the name of the userdb connector

set/get userdbConnectorType - the userdb connector type (valid values for the 'userdb\_connector\_type' are: "local", "ldap" or "ldapBind")

set/get usermapLocalFile - the local file of the usermap (this is used when the user mapping information is local.)

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: Webadmin  
Parents:  
Server->configWebadmin

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the Webadmin service  
set minConnThreads - the minimum number of threads involved in I/O for the Webadmin service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get sessionIdleTimeout - the time interval after which an idle session is terminated  
set/get sessionActivityTimeout - the maximum time interval after which a Webadmin session is terminated even if activity exists  
set/get allowKeepAlive - permanent connections  
set/get allowLargeIncomingData - the data receiving to be allowed after the session is terminated  
set/get httpHeadersMaxSize - the maximum allowed size for received HTTP headers  
set/get httpBodyMaxSize - the maximum allowed size for incoming HTTP body  
set/get uploadMaxSize - the maximum allowed size for incoming upload data  
set/get path - the Webadmin's directory path  
set/get logLevel - the log level for the Webadmin service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the Webadmin service

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the Webadmin configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the Webadmin service

Return: bool

**Method Name: stop**

Description: This function stops the Webadmin service

Return: bool

**Method Name: getAccessControl**

Description: This function returns an access control object for the Webadmin service

Return: AccessControl

**Method Name: addUrlRedirect**

Description: adds an Webadmin URL redirect

Return: bool

Parameters:

string : Address ( the address )

string : Host ( the host )

string : Port ( the port )

**Method Name: removeUrlRedirect**

Description: removes an Webadmin URL redirect

Return: bool

Parameters:

string : Address ( the address )

**Method Name: listUrlRedirects**

Description: list all URL redirects

Return: array

**Method Name: updateUrlRedirect**

Description: updates an Webadmin URL redirect

Return: bool

Parameters:

string : Address ( the address )

string : NewHost ( the new host )

string : NewPort ( the new port )

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

---

Object: Webmail  
Parents:  
Server->configWebmail

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the Webmail service  
set/get maxErrors - the maximum protocol errors number after which the connection is terminated  
set/get maxAuthErrors - the maximum authentication errors number after which the connection is terminated  
set/get sessionIdleTimeout - the time interval after which an idle session is terminated  
set/get sessionActivityTimeout - sets the maximum time interval after which a Webmail session is terminated even if activity exists  
set/get allowKeepAlive - permanent connections  
set/get allowLargeIncomingData - data receiving to be allowed after the session is terminated  
set/get httpHeadersMaxSize - the maximum allowed size for received HTTP headers  
set/get httpBodyMaxSize - the maximum allowed size for incoming HTTP body  
set/get uploadMaxSize - the maximum allowed size for incoming upload data  
set/get path - the Webmail's directory path  
set/get showDomainList - if the domain displaying is enabled  
set/get userdbConnectorType - the userdb connector type  
set/get userdbConnectorName - the userdb connector name  
set/get logLevel - the log level for the Webmail service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the Webmail service  
set/get enableIcalCalendar - enable/disable Calendar (ICS) access through iCal/HTTP  
set/get enableIcalFreebusy - enable/disable Free/Busy (IFB) access through iCal/HTTP  
set/get userType - ajax interface type [<ajaxDefault | ajaxOnly | standardDefault | standardOnly>]

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the Webmail configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the Webmail service

Return: bool

**Method Name: stop**

Description: This function stops the Webmail service

Return: bool

**Method Name: addHostNameResolver**

Description: This function adds a hostname resolver. This maps a hostname to a domain

Return: bool

Parameters:

string : host ( hostname )

string : domain ( domain name )

**Method Name: listHostNameResolvers**

Description: This method lists the hostname resolvers

Return: array

**Method Name: updateHostNameResolver**

Description: This function changes the hostname resolver that matches the given host

Return: bool

Parameters:

string : hostname ( the hostname )

string : newDomain ( the new domain value )

**Method Name: removeHostNameResolver**

Description: This function removes the hostname resolver that matches the given host

Return: bool

Parameters:

string : hostname ( the hostname )

**Method Name: addUriRedirect**

Description: This function adds a URL redirect. This can be used to redirect the request for a specific host to another address.

Return: bool

Parameters:

String : address ( the address ex. 127.0.0.1:8000 )

string : host ( the hostname )

int : port ( the port number )

---

**Method Name: listUrlRedirects**

Description: This method lists the URL redirects

Return: array

---

**Method Name: removeUrlRedirect**

Description: This function removes an URL redirect with the specified address

Return: bool

Parameters:

: address ( )

---

**Method Name: updateUrlRedirect**

Description: This function changes an existing URL redirect, changing its host and port

Return: bool

Parameters:

: address ( )

: newHost ( )

: newPort ( )

---

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

---

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

---

Object: WebmailAdvertising  
Description: this object contains the WebmailAdvertising data  
Parents:  
Account->configWebmailAdvertising  
AccountClass->configWebmailAdvertising  
Domain->configAccountDefaultWebmailAdvertising

**Members:**

set/get enableTopBanner - enabled/disabled top Webmail banner  
set/get enableRightBanner - enabled/disabled right Webmail banner  
set/get enableBottomBanner - enabled/disabled bottom Webmail banner  
set/get topBannerContent - the content of the top banner  
set/get rightBannerContent - the content of the right banner  
set/get bottomBannerContent - the content of the bottom banner

---

**Method Name: isInherited**

Description: check if the variable is inherited

Return: int (1 if the value is inherited, 0 if not, or -1 if this does not apply to the current quota object)

Parameters:

string : VariableName ( the variable )

---

**Method Name: inherit**

Description: sets the variable to inherit from the selected source

Return: bool

Parameters:

string : VariableName ( the variable )

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: WebmailData

Description: this is the object containing the Webmail data for the selected account

Parents:

Account->configWebmailData

MailList->configWebmailData

---

**Members:**

set/get skin - the skin for Webmail

set/get language - the language for Webmail

set/get signature - the account signature

set/get pageSize - page size

set/get timeZone - the timezone of the account

set/get saveToSent - a mail copy to be saved in the Sent folder

set/get deleteToTrash - deleted mail to be sent in the Trash folder

set/get autoRefreshInterval - the Webmail automatic refresh interval, stated in minutes

set/get confirmMailDelete - confirmation of mail deletion

set/get confirmFolderEmpty - confirmation of emptying folder

set/get htmlFilterLevel - the security level for a HTML mail body

set/get usePublicContacts - if the account should use the domain public contacts in addressbook context

set/get useRcptContacts - if the account should use the domain recipient contacts in addressbook context

set/get requestReadReceipts - enable/disable the requesting of read receipts

set/get sendReadReceipts - enable/disable sending of read receipts (values: always|never|ask)

get weekStartDay - the account week start day

get hideDeletedMessages - deleted mail to be sent in the Trash folder

get replyTo - account reply to

get enableNewEmailNotification - the new email notification status

---

**Method Name: save**

Description: this method must be called if one of the members has been changed

Return: bool

---

Object: WebmailProxy  
Parents:  
Server->configWebmailProxy

**Members:**

set/get maxConnThreads - the maximum number of threads involved in I/O for the Webmail service  
set/get minConnThreads - the minimum number of threads involved in I/O for the Webmail service  
set/get maxErrors - the maximum number of protocol errors after which the connection is terminated  
set/get maxAuthErrors - the maximum number of authentication errors after which the connection is terminated  
set/get authenticateOnProxy - if the authentication should be also done on proxy  
set/get path - the WebmailProxy's directory path  
set/get allowKeepAlive - permanent connections  
set/get httpHeadersMaxSize - the maximum allowed size for received HTTP headers  
set/get httpBodyMaxSize - the maximum allowed size for incoming HTTP body  
set/get uploadMaxSize - the maximum allowed size for incoming upload data  
set/get requestsQueueMaxSize - the maximum size of the requests? queue  
set/get allowLargeIncomingData - data receiving to be allowed after the session times out  
set/get enableMobileWebmail - enabled/disabled the Mobile Webmail interface  
set/get mappingDataUsermapName - the mapping usermap name for the proxy  
set/get mappingDataMappingHost - the mapping data host for the proxy  
set/get mappingDataMappingPort - the mapping data port for the proxy  
set/get connectionDataTimeout - the timeout for the read/write operation  
set/get connectionDataMaxConnections - the maximum number of simultaneous connections  
set/get connectionDataSslEnable - if the connection should be SSL  
set/get connectionDataLocalInterface - the address of the interface used to make the connections  
set/get userdbConnectorType - the userdb connector type  
set/get userdbConnectorName - the userdb connector name  
set/get logLevel - the log level for the Webmail service  
set/get logHost - the address of the host that offers log services  
set/get logType - the log type for the Webmail service  
set/get userInterfaceType - ajax interface type [<ajaxDefault | ajaxOnly | standardDefault | standardOnly>]

**Method Name: updateListener**

Description: This function updates a listener of this service

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: configAccessControl**

Description: This function returns an access control object for this service

**Method Name: save**

Description: This function saves the Webmail Proxy configuration back to Axigen

Return: bool

**Method Name: reset**

Description: This function resets the changes made but not yet saved

Return: bool

**Method Name: start**

Description: This function starts the Webmail Proxy service

Return: bool

**Method Name: stop**

Description: This function stops the Webmail Proxy service

Return: bool

**Method Name: addListener**

Description: This function adds a listener to this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: removeListener**

Description: This function removes a listener from this service

Return: bool

Parameters:

string : Address ( the listener address (ex: 127.0.0.1:25) )

**Method Name: listListeners**

Description: This function lists the listeners for this service

Return: array

**Method Name: addHostNameResolver**

Description: This function adds a hostname resolver. This maps a hostname to a domain

Return: bool

Parameters:

string : host ( hostname )

string : domain ( domain name )

---

**Method Name: listHostNameResolvers**

Description: This method lists the hostname resolvers

Return: array

---

**Method Name: updateHostNameResolver**

Description: This function changes the hostname resolver that matches the given host

Return: bool

Parameters:

string : hostname ( the hostname )

string : newDomain ( the new domain value )

---

**Method Name: removeHostNameResolver**

Description: This function removes the hostname resolver that matches the given host

Return: bool

Parameters:

string : hostname ( the hostname )

---

**Method Name: addUrlRedirect**

Description: This function adds a URL redirect. This can be used to redirect request for a specific host to another address.

Return: bool

Parameters:

String : address ( the address ex. 127.0.0.1:8000 )

string : host ( the hostname )

int : port ( the port number )

---

**Method Name: listUrlRedirects**

Description: This method lists the URL redirects

Return: array

---

**Method Name: removeUrlRedirect**

Description: This function removes an URL redirect with the specified address

Return: bool

Parameters:

: address ( )

---

**Method Name: updateUrlRedirect**

Description: This function changes an existing URL redirect, changing its host and port

Return: bool

Parameters:

: address ( )

: newHost ( )

: newPort ( )

---