

# CORE API Reference

## User Management and Accounting

**Version 2.6**  
June 30, 2009

### Proprietary Information

This document is our property. It may be used by recipient only for the purpose for which it was transmitted and shall be returned upon request or when no longer needed by recipient. It may not be copied or communicated without the prior written consent of us.

**COPYRIGHT NOTIFICATION**

Copyright © 2009. All rights reserved.

**DISCLAIMER AND LIMITATION OF LIABILITY**

We made efforts to ensure the accuracy and completeness of all information in this document. However, we make no warranties of any kind (whether express, implied or statutory) with respect to the information herein.

We assume no liability to any party for loss or damage (whether direct or indirect) caused by any errors, omissions or statements of any kind contained in this document. Further, we assume no liability arising from the application or use of the product or service described herein and specifically disclaims any representation that the products or services described herein do not infringe upon any existing or future intellectual property rights. Nothing herein grants the reader any license to make, use, or sell equipment or products constructed in accordance with this document. Finally, all rights and privileges related to any intellectual property right described herein are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner.

We reserve the right to make changes to any information herein without further notice.

**NOTICE AND CAUTION concerning Patent or Trademark Rights**

The inclusion in this document, the associated on-line file, or the associated software of any information covered by any patent, trademark, or service mark rights shall not constitute nor imply a grant of, or authority to exercise, any right or privilege protected by such patent, trademark, or service mark. All such rights and privileges are vested in the patent, trademark, or service mark owner, and no other person may exercise such rights without express permission, authority, or license secured from the patent, trademark, or service mark owner.

This publication was created using OpenOffice.org by Sun Microsystems Inc.

**CORE API Reference**  
Version 2.6

June 30, 2009

## TABLE OF CONTENTS

1. Introduction.....	4
2. User Relations.....	5
3. The Environment Database.....	7
4. Commands.....	9
4.1 User management commands.....	10
4.1.1 CheckUsername.....	10
4.1.2 AddUser.....	12
4.1.3 ModifyUser.....	14
4.1.4 AssignUser.....	16
4.1.5 StatusUser.....	17
4.1.6 DeleteUser.....	19
4.1.7 FindUser.....	20
4.1.8 QueryUserList.....	22
4.1.9 QueryUserObjectStatistics.....	24
4.1.10 QueryObjectUserList.....	25
4.1.11 GetUserBranch.....	26
4.1.12 StatusUserclass.....	28
4.1.13 ModifyUserClass.....	30
4.1.14 QueryUserClassList.....	31
4.1.15 ModifyUserPassword.....	32
4.2 User role management commands (ACL).....	33
4.2.1 QueryRoleUserList.....	33
4.2.2 CreateRoleUser.....	34
4.2.3 StatusRoleUser.....	36
4.2.4 UpdateRoleUser.....	37
4.2.5 DeleteRoleUser.....	38
4.3 Environment management commands.....	39
4.3.1 GetEnvironment.....	39
4.3.2 SetEnvironment.....	41
4.3.3 QueryEnvironmentList.....	42
4.4 Event management commands.....	44
4.4.1 StatusEvent.....	44
4.4.2 DeleteEvent.....	46
4.4.3 QueryEventList.....	47
4.5 Accounting commands.....	49
4.5.1 AddAccounting.....	49
4.5.2 ModifyAccounting.....	51
4.5.3 StatusAccounting.....	53
4.5.4 DeleteAccounting.....	54
4.5.5 ExecutePayment.....	55
4.5.6 QueryAccountingList.....	56
4.6 Miscellaneous commands.....	59
4.6.1 CheckAuthentication.....	59
4.6.2 CheckAuthorization.....	60
4.6.3 AssignObject.....	62
4.6.4 SetObjectConfigClass.....	64
4.6.5 GetObjectConfigClass.....	65
4.6.6 QueryObjectList.....	66
4.6.7 QueryExchangeRates.....	68
4.6.8 QueryObjectRenewalList.....	70
4.6.9 QueryObjectLogList.....	72
4.6.10 QueryObjectDeletionLogList.....	74
4.6.11 StatusObjectLog.....	75
4.6.12 StatusObjectDeletionLog.....	77
5. Additional Services.....	78
5.1 POP3-Event-Server.....	78
6. System response code list.....	79
7. ChangeLog.....	81

# 1. Introduction

This reference manual describes the API services available to all users, it is however not an programming tutorial itself.

The manual arranges the commands into the following order:

- User management commands,
- Environment management commands,
- Event management commands,
- Accounting commands and
- Miscellaneous commands.

The API itself is not object orientated, but object related and can be encapsulated into any OO language without much hassle.

If there are any errors or missing topics, please don't hesitate to contact us!

## 2. User Relations

The relations define the operational parameters of a user and can only be modified by its superordinate users. The most important use of relations is to defined the prices and limits.

For example the system supports the following relations for domains:

Relation type	Description
ZONES	A comma separated list of zones the user has access to, e.g. ZONES:com,net,org,info,biz,cn,com.cn,net.cn,org.cn
PRICE_CLASS_DOMAIN_DOMAINCLASS_ANNUAL	The default annual fee for domains of the class <i>DOMAINCLASS</i> , e.g. PRICE_CLASS_DOMAIN_COM_ANNUAL:12.00
PRICE_CLASS_DOMAIN_DOMAINCLASS_ANNUALPERIOD	The annual fee for domains of the class <i>DOMAINCLASS</i> if period is <i>PERIOD</i> , e.g. PRICE_CLASS_DOMAIN_COM_ANNUAL5:10.95
PRICE_CLASS_DOMAIN_DOMAINCLASS_SETUP	The setup fee for domains of the class <i>DOMAINCLASS</i> , e.g. PRICE_CLASS_DOMAIN_COM_SETUP:0.00
PRICE_CLASS_DOMAIN_DOMAINCLASS_TRANSFER	The transfer fee for domains of the class <i>DOMAINCLASS</i> , e.g. PRICE_CLASS_DOMAIN_COM_TRANSFER:12.00 (.com Domains are renewed for one year if transferred)
PRICE_CLASS_DOMAIN_DOMAINCLASS_TRANSFERAPPLICATION	The fee for transfer application is a part of the transfer fee which will not be refunded by the registry in case of a failed transfer. Some registries (like DNS.LU) have a non-refundable fee like this. So if a user starts a transfer he always will get charged for both the transfer fee and the transfer application fee, but will only get back the transfer fee if the transfer fails.
PRICE_CLASS_DOMAIN_DOMAINCLASS_RENEWPERIOD	This relation can be used in case that the renewal price for a domain differs from the annual price. This could be interesting if you start some kind of promotion. You also have the possibility to define periods in which the renewal is cheaper than in others (e.g. 8.00 USD if you renew a domain for 1 or 2 years, but only 6.00 USD if you renew a domain for 3 or more years).
PRICE_CLASS_DOMAIN_DOMAINCLASS_CURRENCY	The currency of relations for domains of the class <i>DOMAINCLASS</i> , e.g. PRICE_CLASS_DOMAIN_COM_CURRENCY:USD

The possible domain classes are defined below. If there is no currency for a domain class, then the system uses the users account currency instead.

**Prices are calculated according to the daily exchange rate of the ECB (European Central Bank, [www.ecb.int](http://www.ecb.int)), if the domain class currency and user account currency are different!**

The relation parameters for AddUser and ModifyUser are build using the following syntax:

relationN = RELATIONTYPE:RELATIONVALUE

e. g.:

```

relation0 = ZONES:com,net,org,de
relation1 = PRICE_CLASS_DOMAIN_COM_ANNUAL:12.00
relation2 = PRICE_CLASS_DOMAIN_COM_SETUP:0.00
relation3 = PRICE_CLASS_DOMAIN_COM_TRANSFER:12.00
relation4 = PRICE_CLASS_DOMAIN_NET_ANNUAL:12.00
relation5 = PRICE_CLASS_DOMAIN_NET_SETUP:0.00
relation6 = PRICE_CLASS_DOMAIN_NET_TRANSFER:12.00
relation7 = PRICE_CLASS_DOMAIN_ORG_ANNUAL:12.00
relation8 = PRICE_CLASS_DOMAIN_ORG_SETUP:0.00
relation9 = PRICE_CLASS_DOMAIN_ORG_TRANSFER:12.00
relation10 = PRICE_CLASS_DOMAIN_DE_ANNUAL:6.00
relation11 = PRICE_CLASS_DOMAIN_DE_SETUP:3.00
relation12 = PRICE_CLASS_DOMAIN_DE_TRANSFER:6.00

```

Implementations should manage relation tables as text files and pass them as an array of lines to the libraries. The text files should look like that (e. g.):

```
ZONES:com,net,org,de
PRICE_CLASS_DOMAIN_COM_ANNUAL:12.00
PRICE_CLASS_DOMAIN_COM_SETUP:0.00
PRICE_CLASS_DOMAIN_COM_TRANSFER:12.00
PRICE_CLASS_DOMAIN_NET_ANNUAL:12.00
PRICE_CLASS_DOMAIN_NET_SETUP:0.00
PRICE_CLASS_DOMAIN_NET_TRANSFER:12.00
PRICE_CLASS_DOMAIN_ORG_ANNUAL:12.00
PRICE_CLASS_DOMAIN_ORG_SETUP:0.00
PRICE_CLASS_DOMAIN_ORG_TRANSFER:12.00
PRICE_CLASS_DOMAIN_DE_ANNUAL:6.00
PRICE_CLASS_DOMAIN_DE_SETUP:3.00
PRICE_CLASS_DOMAIN_DE_TRANSFER:6.00
```

This makes it very easy to manage different levels of user accounts and pricing plans.

### 3. The Environment Database

Every user within the system has an associated Environment Database.

The environment is used to store user specific data like an address, preferences and so on within a variable tree structure.

There are three commands dedicated to access the environment:

- **GetEnvironment**,
- **SetEnvironment** and
- **QueryEnvironmentList**.

Furthermore the **AddUser** and **ModifyUser** commands modify the environment db of subusers.

There is no fixed or required structure for the Environment DB, however implementations should use the same structure for the most common elements:

Environment Key/Name	Description
user-info/contact/company/organization	Organization of the user
user-info/contact/company/address1	Street address
user-info/contact/company/address2	2 <sup>nd</sup> line of the street address (optional)
user-info/contact/company/city	City
user-info/contact/company/state	State (optional)
user-info/contact/company/zip	ZIP / Postal-Code
user-info/contact/company/country	2-letter ISO-3166 Country code, e. g. DE, US...
user-info/contact/company/title	Title of the primary contact person (optional)
user-info/contact/company/firstname	First name of the primary contact person
user-info/contact/company/lastname	Last name of the primary contact person
user-info/contact/company/phone	Phone number, international format
user-info/contact/company/fax	Fax number, international format (optional)
user-info/contact/company/email	Email address
user-info/contact/company/email2	Secondary email address (optional)
user-info/contact/company/vatid	V.A.T.-I.D. (optional), e. g. DE123456789
user-info/contact/company/billingtype	Billing type: debit banktransfer
user-info/contact/company/accountnumber	For billing type "debit": accountnumber
user-info/contact/company/accontholder	For billing type "debit": accontholder
user-info/contact/company/bankcode	For billing type "debit": bankcode
user-info/contact/company/bankname	For billing type "debit": bankname
user-info/contact/company/debitstatus	For billing type "debit": debit status active inactive

Implementations may not use keys starting with SYSTEM/ for random data, as they are reserved for use by the system.

The environment parameters for AddUser and ModifyUser are build using the following syntax:

*environmentN = ENVIRONMENTKEY/NAME:VALUE*

e. g.:

```
environment0 = user-info/contact/company/firstname:John
environment1 = user-info/contact/company/lastname:Doe
environment2 = user-info/contact/company/organization:Domain Reseller Inc.
...
```

## 4. Commands

All requests are submitted using the following HTTP interface:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Replace the **BASE-URL** placeholder by the appropriate URL for the OT&E or Production environment. There is an optional parameter “**s\_user**” that defines the user as which the command should be executed. This user must be the same as the login user or one of its subusers.

The server responds with “Content-type: text/plain”, using the following syntax:

```
[RESPONSE]  
code = 200  
description = Command completed successfully  
property[propertyname1][0] = value1  
property[propertyname1][1] = value2  
property[propertyname1][2] = value3  
...  
property[propertyname2][0] = ...  
EOF
```

Implementations should handle the responses case-insensitive. All spaces within property names must be ignored, e. g. “created date” should be read as “CREATEDDATE”, “registration expiration date” as “REGISTRATIONEXPIRATIONDATE” and so on.

Implementations **MUST** verify that all responses end with “EOF”.

Date properties are returned in SQL syntax (e. g. “2003-09-01 12:00:00”) using the timezone UTC.

## 4.1 User management commands

The following commands are used to create, manage and remove subuser accounts.

### 4.1.1 CheckUsername

#### Description

Checks if a user id is valid and available.

#### Availability

All users have access to this command.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	User ID of the subuser. The ID must have the same syntax as domain names, e.g. <b>subuser.com</b>	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully – The subuser ID is still available
505	The syntax of the subuser ID is invalid
540	The subuser ID is not available
541	Missing Parameter

Property	min	max	Description
no properties are returned			

#### Notes

- All subuser IDs must look like domain names, e.g. like myname.com

#### Example

The following request checks the subuser id “subreseller.com”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=CheckUser
&subuser=subreseller.com
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

## 4.1.2 AddUser

### Description

Create a new subuser.

### Availability

Your account must have the **ALLOW\_SUBUSER** relation set to 1.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	New user ID	TEXT
<b>password</b>	required	Password for the new user	TEXT
<b>credit</b>	optional	Credit limit of the new user	DECIMAL(10,2)
<b>vat</b>	required	Additional V.A.T. % for the new user, e. g. 16.00	DECIMAL(10,2)
<b>currency</b>	required	Currency of the new user account, e. g. USD	TEXT
<b>relationN</b>	optional	Relations for the new user	RELATION [0..N]
<b>environmentN</b>	optional	Environment for the new user	TEXT [0..N]
<b>userclass</b>	optional	Name of a UserClass (see ModifyUserClass to learn more about UserClasses)	TEXT
<b>active</b>	optional	Activates/deactivates the user account	INT (0   1)

### Returned properties and values

Code	Description
200	Command completed successfully
541	One of the parameters was invalid

Property	min	max	Description
no properties are returned			

## Example

The following request creates a new subuser “subreseller.com”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=AddUser
&subuser=subreseller.com
&password=topsecret
&credit=0.00
&currency=EUR
&vat=16.00
&relation0=ZONES:com,net
&relation1=PRICE_CLASS_DOMAIN_COM_ANNUAL:9.00
&relation2=PRICE_CLASS_DOMAIN_COM_SETUP:0.00
&relation3=PRICE_CLASS_DOMAIN_NET_ANNUAL:9.00
&relation4=PRICE_CLASS_DOMAIN_NET_SETUP:0.00
&environment0=user-info/contact/company/organization:Testcompany+GmbH
&environment1=user-info/contact/company/street:Teststrasse+123
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

### 4.1.3 ModifyUser

#### Description

Modify an existing subuser.

#### Availability

All users have access to this command.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	Subuser ID	TEXT
<b>newsuser</b>	optional	New Subuser ID	TEXT
<b>password</b>	optional	New password for the subuser	TEXT
<b>credit</b>	optional	Credit limit of the subuser	DECIMAL(10,2)
<b>vat</b>	optional	Additional V.A.T. % for the subuser, e. g. 16.00	DECIMAL(10,2)
<b>currency</b>	optional	Currency of the subuser account, e. g. USD	TEXT
<b>relationN</b>	optional	Relations for the subuser	RELATION [0..M]
<b>environmentN</b>	optional	Environment for the subuser	TEXT [0..M]
<b>userclass</b>	optional	Name of a UserClass (see ModifyUserClass to learn more about UserClasses)	TEXT
<b>active</b>	optional	Activates/deactivates the user account	INT (0   1)

#### Returned properties and values

Code	Description
200	Command completed successfully
541	One of the parameters was invalid

Property	min	max	Description
no properties are returned			

#### Notes

- New relation values overwrite existing relations with the same name
- Empty relation values delete existing relations with the same name
- New environment values overwrite existing environment values with the same name
- Empty environment values delete existing environment values with the same name

## Example

The following request modifies the subuser “subreseller.com”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=ModifyUser
&subuser=subreseller.com
&password=topsecret
&credit=1000.00
&relation0=ZONES:com,net,org
&relation1=PRICE_CLASS_DOMAIN_ORG_ANNUAL:9.00
&relation2=PRICE_CLASS_DOMAIN_ORG_SETUP=0.00
&relation3=PRICE_CLASS_DOMAIN_NET_ANNUAL:10.00
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

## 4.1.4 AssignUser

### Description

Assign an existing subuser to another parent user.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	Subuser ID	TEXT
<b>user</b>	optional	New Parentuser ID	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	One of the parameters was invalid

Property	min	max	Description
no properties are returned			

### Notes

- Subusers can only be moved under your user account.
- All objects, environment entries, accounting records and invoices are assigned to the subuser and are moved with the subuser.
- To move objects see API function "AssignObject".

## 4.1.5 StatusUser

### Description

Query the status properties and relations of a user.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
subuser	optional	User ID of a subuser	TEXT
stats	optional	On value "1", StatusUser also shows "subusers direct" and "subusers total"	INTEGER (0   1)

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
USER	1	1	The user ID
PARENTUSER	1	1	The user ID of the parent user
ACCOUNTCURRENT	1	1	The current account balance
ACCOUNTCREDIT	1	1	The credit limit for this user
ACCOUNTVAT	1	1	The VAT for this account in percent (%), e. g. 16.00
ACCOUNTCURRENCY	1	1	The currency of this account, e. g. USD
SUBUSERSDIRECT	1	1	The number of direct subusers
SUBUSERSTOTAL	1	1	The number of all subusers
RELATIONTYPE	0	N	Type of the <i>n</i> -th user relation
RELATIONVALUE	0	N	Value of the <i>n</i> -th user relation
USERCLASS	0	1	The name of the UserClass

### Example

The following request queries the user status of "reseller.de":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=StatusUser
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[user][0] = subreseller.com
property[parent user][0] = reseller.de
property[account current][0] = 10000.00
property[account credit][0] = 0.00
property[account vat][0] = 16.00
property[account currency][0] = USD
property[subusers direct][0] = 0
property[subusers total][0] = 0
property[relation type][0] = PRICE_CLASS_DOMAIN_COM_ANNUAL
property[relation value][0] = 19.95
property[relation type][1] = PRICE_CLASS_DOMAIN_COM_CURRENCY
property[relation value][1] = USD
property[relation type][2] = PRICE_CLASS_DOMAIN_COM_SETUP
property[relation value][2] = 0.00
property[relation type][3] = PRICE_CLASS_DOMAIN_COM_TRANSFER
property[relation value][3] = 19.95
property[relation type][4] = PRICE_CLASS_DOMAIN_DE_ANNUAL
property[relation value][4] = 6.00
property[relation type][5] = PRICE_CLASS_DOMAIN_DE_CURRENCY
property[relation value][5] = EUR
property[relation type][6] = PRICE_CLASS_DOMAIN_DE_SETUP
property[relation value][6] = 3.00
property[relation type][7] = PRICE_CLASS_DOMAIN_DE_TRANSFER
property[relation value][7] = 6.00
property[relation type][8] = PRICE_CLASS_DOMAIN_NET_ANNUAL
property[relation value][8] = 35.00
property[relation type][9] = PRICE_CLASS_DOMAIN_NET_CURRENCY
property[relation value][9] = USD
property[relation type][10] = PRICE_CLASS_DOMAIN_NET_SETUP
property[relation value][10] = 0.00
property[relation type][11] = PRICE_CLASS_DOMAIN_NET_TRANSFER
property[relation value][11] = 35.00
property[relation type][12] = PRICE_CLASS_DOMAIN_ORG_ANNUAL
property[relation value][12] = 35.00
property[relation type][13] = PRICE_CLASS_DOMAIN_ORG_CURRENCY
property[relation value][13] = USD
property[relation type][14] = PRICE_CLASS_DOMAIN_ORG_SETUP
property[relation value][14] = 0.00
property[relation type][15] = PRICE_CLASS_DOMAIN_ORG_TRANSFER
property[relation value][15] = 35.00
property[relation type][16] = ZONES
property[relation value][16] = com,net,org,biz,info,de
EOF
```

## 4.1.6 DeleteUser

### Description

Deletes a subuser.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
subuser	required	User ID of the subuser.	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The subuser could not be deleted

Property	min	max	Description
no properties are returned			

### Notes

- Only own subusers can be deleted
- The subuser must not have any objects in its account
- The subuser must not have any subusers itself
- The account of the subuser must have a balance of 0.00

### Example

The following request deletes the subuser "subreseller.com":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command>DeleteUser
&subuser=subreseller.com
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

## 4.1.7 FindUser

### Description

Finds a certain user by searching for a certain keyword through the user's environments "user-info/contact/company" and "user-info/contact/private". Additionally you may use FindUser to search for an exact username or a part of such one.

### Availability

All users have access to this command. The command only works if you are searching for your own subusers.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>userdepth</b>	optional	Depth of the list, may be SELF, SUBUSER or ALL	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>pattern</b>	required	search pattern	TEXT
<b>exactmatch</b>	optional	flag to control if the search should match exactly or not; can be 0 or 1, DEFAULT: 0	TEXT
<b>matchmode</b>	optional	USER: search for username ENVIRONMENT: search through the enviroments mentioned above Default value: USER	TEXT (USER   ENVIRONMENT)
<b>invert</b>	optional	If invert = 1, all users are found which DO NOT include the search pattern	INT (0   1)

**Returned properties and values**

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
USER	1	N	The user ID
PARENTUSER	1	N	The user ID of the parent user
ENVIRONMENTKEY	-	-	The environment key where the search pattern has been found
ENVIRONMENTNAME	-	-	The environment name where the search pattern has been found
ENVIRONMENTVALUE	-	-	The environment value where the search pattern has been found
FIRST	1	N	The index of the first entry
LAST	1	N	The index of the last entry
LIMIT	1	N	
COUNT	1	N	The number of entries returned with the restrictions you gave.
TOTAL	1	N	The number of entries returned in total

## 4.1.8 QueryUserList

### Description

Query a list of subusers.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>first</b>	optional	Index of the first entry to be returned	INTEGER
<b>limit</b>	optional	Max. number of entries to be returned	INTEGER
<b>userdepth</b>	optional	Depth of the list, may be SELF, SUBUSER or ALL	TEXT
<b>userN</b>	optional	select specific users as SELF or as SUBUSER to pick up different information	TEXT
<b>userdepthN</b>	optional		TEXT (SELF SUBUSER)
<b>wide</b>	optional	if wide is set to 1, a detailed list with all possible entities from the given user	INTEGER (0 1)
<b>active</b>	optional	List shows only active/inactive users	INT (0   1)
<b>subuser</b>	optional	List shows only users of given subuser	TEXT
<b>mincreateddate</b>	optional	Search for minimum created date of user	DATE
<b>maxcreateddate</b>	optional	Search for maximum created date of user	DATE
<b>minupdateddate</b>	optional	Search for minimum updated date of user	DATE
<b>maxupdateddate</b>	optional	Search for maximum updated date of user	DATE
<b>userclass</b>	optional	Search for certain user class	TEXT
<b>accountvat</b>	optional	Search for certain account VAT, e.g. 19.00	INTEGER
<b>orderby</b>	optional	Order of the list: CREATEDDATE and CREATEDDATEDESC	TEXT

## Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
USER	1	N	The user ID
PARENTUSER	1	N	The user ID of the parent user
COUNT	1	1	The number of entries returned
FIRST	1	1	The index of the first entry
LAST	1	1	The index of the last entry
LIMIT	1	1	The query limit (max. number of entries returned)
TOTAL	1	1	The total number of entries found

## Notes

- userdepth defines the depth of the returned list. SELF returns a list with only one element, the user itself. SUBUSER returns a list with all direct subusers. ALL returns a list with all subusers and all subusers below them.

## Example

The following request queries all direct subusers of “reseller.de”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=QueryUserList
&userdepth=SUBUSER
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[user][0] = subreseller.com
property[parent user][0] = reseller.de
property[user][1] = subuser.com
property[parent user][1] = reseller.de
property[user][2] = domainseller.de
property[parent user][2] = reseller.de
property[count][0] = 3
property[first][0] = 0
property[last][0] = 2
property[limit][0] = 1000
property[total][0] = 3
EOF
```

## 4.1.9 QueryUserObjectStatistics

### Description

Query a set of statistics to a certain user account, such as number of domains, number of rootservers, number of mobile accounts or number of SSL certificates.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	Subuser of which the statistics want to be queried	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed
531	Authorization failed

Property	min	max	Description
DOMAIN	1	N	Number of domains assigned to the subuser
CONTACT	1	N	Number of contact handles assigned to the subuser
DNSZONE	1	N	Number of DNS zones assigned to the subuser
SSLCERT	1	N	Number of SSL certificates assigned to the subuser
MOBILEACCOUNT	1	N	Number of mobile accounts assigned to the subuser
VSERVER	1	N	Number of rootservers assigned to the subuser
SERVICEPHONE	1	N	Number of service phone IPs assigned to the subuser

### Notes

- The command can only be executed for own subusers. If you try to query the statistics of an other subuser, the command will fail with error code 531 (authorization failed).

### 4.1.10 QueryObjectUserList

#### Description

Queries the usernames related to the array of objects which has been given to the command.

#### Availability

All users have access to this command.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret&command=command&parameter1=value1&parameter2=value2&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>level</b>	optional	PEER   ENDUSER (default) supplies either the user underneath the one who executed the command' (peer) or the enduser related to the object (enduser)	TEXT
<b>ObjectidN</b>	required	Array of object Ids	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

### 4.1.11 GetUserBranch

#### Description

Query the hierarchical list of subusers between your user and the given subuser

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
subuser	required	User ID of the subuser to get the branch to	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
USER	1	N	The user ID

#### Example

The following request queries the subusers between you and "reseller.de":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=GetUserBranch
&subuser=reseller.de
```

The user hierarchy is as follows:

```
myuser.de
|-- subreseller.com
|   |-- subuser.com
|       |-- reseller.de
|       |-- ...
|   |-- ...
|   ...
|-- ...
...
...
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[user][0] = subreseller.com
property[user][1] = subuser.com
property[user][2] = reseller.de
EOF
```

## 4.1.12 StatusUserclass

### Description

Query information about UserClass.

A UserClass can be used to group relations: Users that have the same conditions, have the same UserClass. Any changes on the UserClass relations affect all users that have this UserClass. UserClass can be set with the "userclass" parameter at commands AddUser and ModifyUser. The parent users of the user see the userclass at command StatusUser, the user itself only see the relations produced by the UserClass and not the name of the UserClass.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
userclass	required	Name of the user class	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

### Example

The following request queries informations about the userclass "userclass":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=StatusUserclass
&userclass=userclass
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[relation type][0] = DNSZONE_EXTERNAL_RRS_INCLUDED
property[relation value][0] = 10
property[relation type][1] = DNSZONE_INTERNAL_RRS_INCLUDED
property[relation value][1] = 10
property[relation type][2] = PRICE_CLASS_DNSZONE_EXTERNAL_ANNUAL
property[relation value][2] = 0.30
property[relation type][3] = PRICE_CLASS_DNSZONE_EXTERNAL_CURRENCY
property[relation value][3] = EUR
...
property[relation type][215] = SERVICE_USAGE_SYSTEM_HARDQUOTA
property[relation value][215] = 500
property[relation type][216] = VSERVER_DEBIAN_5GB_TRAFFIC_INCLUDED_GB
property[relation value][216] = 15
property[relation type][217] = ZONES
property[relation value][217] =
ag,at,be,biz,biz.pl,bz,cc,cd,cn,co.at,co.uk,com,com.cn,com.pl,com.sg,com.tw,de,fm,gs,id
v.tw,info,info.pl,jp,la,me.uk,ms,name,
net,net.cn,net.pl,nu,or.at,org,org.cn,org.pl,org.tw,org.uk,pl,se,tc,tk,tv,us,vg,ws

EOF
```

### 4.1.13 ModifyUserClass

#### Description

Create, modify or delete a UserClass. If the UserClass name does not exist, the UserClass is created. If the UserClass has no relations, it is deleted.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>userclass</b>	required	Name of the user class	TEXT
<b>relationN</b>	optional	Relations for the user class	RELATION [0..N]
<b>flushrelations</b>	optional	If set to "1": Delete all existing relations	INTEGER

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

#### 4.1.14 QueryUserClassList

##### Description

Query the list of UserClasses of the current user.

##### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

##### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret&command=command&parameter1=value1&parameter2=value2&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT

##### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
USERCLASS	1	N	The UserClass name

### 4.1.15 ModifyUserPassword

#### Description

Change or reset the password of your own account.

#### Availability

All users have access to this command.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>password</b>	Required	New password	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

## 4.2 User role management commands (ACL)

The following commands are used to manage the user role accounts. User roles can be used to control the access of users to certain API commands.

### 4.2.1 QueryRoleUserList

#### Description

Shows a list of all role users within the current user account.

#### Availability

All users have access to this command.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>userdepth</b>	optional	depth of subuser accounts to check; can be SELF, ALL or SUBUSER	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
ROLE ID	1	N	The role user ID
CREATED DATE	1	1	Date of creation of the role user
UPDATED DATE	1	1	Date of last modification of the role user
DESCRIPTION	1	1	Description entered at the creation of the role user
DEFAULT POLICY	1	1	Default access policy of the role user
STATUS	1	1	Current status of the role user

## 4.2.2 CreateRoleUser

### Description

Creates a role user with certain rights to access various API commands.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>roleid</b>	required	The role user ID	TEXT
<b>password</b>	required	The role user password	TEXT
<b>description0/1</b>	optional	Description text to describe e.g. the position or rights of a user within a company	TEXT
<b>acl1</b>	required	Enter a API command, followed by brackets, a colon, and "ALLOW" or "DENY". Example: RenewDomain():DENY	TEXT
<b>aclN</b>	optional	You may also define ALLOW or DENY only for certain command parts. Example:  acl1 = RenewDomain( acl2 = domain=*.uk acl3=):DENY  Meaning: RenewDomain for .UK domains will get denied	TEXT
<b>status</b>	required	Set the status of the role user to ACTIVE or INACTIVE	TEXT
<b>defaultpolicy</b>	required	If no rights for a certain command are defined, this policy is used	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

## Example

The following request creates a role user “test” with certain rights:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=CreateRoleUser
&roleid=test
&password=topsecret
&description0=Test User
&description1=Please do not use this user
&acl0=QueryDomainList():DENY
&acl1=QueryExtendedDomainList():DENY
&acl2>DeleteDomain():DENY
&acl3=RenewDomain(
&acl4=domain = *.uk
&acl5=period = 1
&acl6=):DENY
&acl7=RenewDomain():ALLOW
&acl8=(domain=somedomainname.):DENY
&status=INACTIVE
&defaultpolicy=ALLOW
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
```

## Explanation

The created user with the role ID “test” can

- generally execute all API commands (*defaultpolicy=ALLOW*)
- generally execute the command RenewDomain (*acl7=RenewDomain():ALLOW*),

but he is not able to

- execute the command QueryDomainList (*acl0=QueryDomainList():DENY*)
- execute the command QueryExtendedDomainList (*QueryExtendedDomainList():DENY*)
- execute the command DeleteDomain (*DeleteDomain():DENY*)
- perform a renewal for a .UK domain

```
(acl3=RenewDomain(
&acl4=domain = *.uk
&acl5=period = 1
&acl6=):DENY)
```

### 4.2.3 StatusRoleUser

#### Description

Using this command, you can query all parameters assigned to a certain role user.

#### Availability

All users have access to this command.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>roleid</b>	required	The role ID of the role user	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
ACL	1	N	The rights of the role user
CREATED DATE	1	1	Date of creation of the role user
UPDATED DATE	1	1	Date of last modification of the role user
DESCRIPTION	1	2	Description entered at the creation of the role user
DEFAULT POLICY	1	1	Default access policy of the role user
STATUS	1	1	Current status of the role user
PASSWORD	1	1	Password of the role user

## 4.2.4 UpdateRoleUser

### Description

This command updates several parameters of a certain role user.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>roleid</b>	required	The role user ID	TEXT
<b>password</b>	optional	The role user password	TEXT
<b>description0/1</b>	optional	Description text to describe e.g. the position or rights of a user within a company	TEXT
<b>acl1</b>	optional	Enter a API command, followed by brackets, a colon, and "ALLOW" or "DENY". Example: RenewDomain():DENY	TEXT
<b>aclN</b>	optional	You may also define ALLOW or DENY only for certain command parts. Example:  acl1 = RenewDomain( acl2 = domain=*.uk acl3=):DENY  Meaning: RenewDomain for .UK domains will get denied	TEXT
<b>status</b>	optional	Set the status of the role user to ACTIVE or INACTIVE	TEXT
<b>defaultpolicy</b>	optional	If no rights for a certain command are defined, this policy is used	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

## 4.2.5 DeleteRoleUser

### Description

With this command you can delete a certain role user.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>roleid</b>	required	The role ID of the role user	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

## 4.3 Environment management commands

The following commands are used to manage the user environment database.

### 4.3.1 GetEnvironment

#### Description

Returns an environment value.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
environmentkey	optional	Key of the environment name (prefix of the name)	TEXT
environmentname	required	Name of the environment value	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed, e. g. because this environment key/name does not exist

Property	min	max	Description
ENVIRONMENTVALUE	1	1	Environment value
PRIVATE	1	1	Is this value hidden for subusers

#### Example

The following request queries the value of the environment “user-info/contact/company/organization”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=GetEnvironment
&environmentkey=user-info/contact/company
&environmentname=organization
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[environmentvalue][0] = Registrar Company Ltd.
property[private][0] = 0
EOF
```

### 4.3.2 SetEnvironment

#### Description

Set an environment value.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
environmentkey	optional	Key of the environment name (prefix of the name)	TEXT
environmentname	required	Name of the environment value	TEXT
environmentvalue	optional	Value of the environmentkey/name	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

#### Notes

- If environmentvalue is not defined, then the request deletes the environmentkey/name

#### Example

The following request sets the value of the environment “user-info/contact/company/organization”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=SetEnvironment
&environmentkey=user-info/contact/company
&environmentname=organization
&value=Testcompany+Inc.
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

### 4.3.3 QueryEnvironmentList

#### Description

Query the list of environment keys or query all name-value pairs for an environmentkey.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
environmentkey	optional	Key of the environment	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed, e. g. because this environment key/name does not exist

Property	min	max	Description
ENVIRONMENTKEY	1	N	Environment key
ENVIRONMENTNAME	1	N	Name of the environment value
ENVIRONMENTVALUE	1	N	Value of the environmentkey/name
PRIVATE	1	N	Is this value hidden for subusers

#### Notes

- If environmentkey is defined, then the request returns all name/value pairs for this key
- If environmentkey is undefined, then the request returns all environmentkeys

#### Example

The following request queries a list of all environmentkeys:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=QueryEnvironmentList
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[environmentkey][0] = user-info/contact/company
property[environmentkey][1] = user-info/contact/billing
property[environmentkey][2] = SYSTEM/events
EOF
```

## 4.4 Event management commands

The following commands are used to query and delete events.

### 4.4.1 StatusEvent

#### Description

Query an event.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
event	required	ID of the event	INTEGER

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
DATE	1	1	Creation date of the event
CLASS	1	1	Class of the event
SUBCLASS	1	1	Subclass of the event
DATA	1	N	Data of the event
INFO	0	1	Additional urlencoded debug info

#### Example

The following request queries an event:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=StatusEvent
&event=12
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2003-09-01 12:00:00.0
property[class][0] = DOMAIN_TRANSFER
property[subclass][0] = TRANSFER_SUCCESSFUL
property[data][0] = domain:testdomain.com
property[data][1] = old registrar:SOMEONE
property[data][2] = new registrar:REGISTRAR INC.
EOF
```

## 4.4.2 DeleteEvent

### Description

Delete an event.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret&command=command&parameter1=value1&parameter2=value2&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>event</b>	required	ID of the event	INTEGER

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

### Example

The following request deletes an event:

```
BASE-URL?s_login=reseller.de&s_pw=secret&command>DeleteEvent&event=12
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

### 4.4.3 QueryEventList

#### Description

Query a list of events.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>first</b>	optional	Index of the first entry to be returned	INTEGER
<b>limit</b>	optional	Max. number of entries to be returned	INTEGER
<b>wide</b>	optional	Returns some more properties if set to 1	INTEGER
<b>class</b>	optional	Event class, e. g. DOMAIN_TRANSFER	TEXT
<b>subclass</b>	optional	Event subclass, e. g. TRANSFER_SUCCESSFUL	TEXT
<b>orderby</b>	optional	Order response by response parameters; switch order by adding "DESC" to response parameter, e.g.: EVENTDATE or EVENTDATEDESC	TEXT

## Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
EVENT	1	N	The event ID
EVENTCLASS	1	N	The event class (returned if wide=1)
EVENTCLASS	1	N	The event subclass (returned if wide=1)
EVENTDATE	1	N	The event date (returned if wide=1)
EVENTDATA0	1	N	Line 1 of the event data (returned if wide=1)
EVENTDATA1	1	N	Line 2 of the event data (returned if wide=1)
EVENTDATA2	1	N	Line 3 of the event data (returned if wide=1)
EVENTDATA3	1	N	Line 4 of the event data (returned if wide=1)
COUNT	1	1	The number of entries returned
FIRST	1	1	The index of the first entry
LAST	1	1	The index of the last entry
LIMIT	1	1	The query limit (max. number of entries returned)
TOTAL	1	1	The total number of entries found

## Notes

- first and limit should be used to implement paging
- Events should be deleted after processing

## Example

The following request queries all events IDs of “reseller.de”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=QueryEventList
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[event][0] = 17
property[event][1] = 18
property[event][2] = 19
property[event][3] = 20
property[count][0] = 4
property[first][0] = 0
property[last][0] = 3
property[limit][0] = 1000
property[total][0] = 4
EOF
```

## 4.5 Accounting commands

The following commands are used to query and manage user accounting tables.

### 4.5.1 AddAccounting

#### Description

Add an accounting into the accounting list of a subuser and update its current account balance.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	ID of the subuser account	TEXT
<b>date</b>	optional	Date of the payment, e. g. 2003-01-01 12:00:00	TEXT
<b>description</b>	required	Description of this accounting entry	TEXT
<b>type</b>	required	Type of this accounting entry, e. g. PAYMENT	TEXT
<b>amount</b>	optional	Amount, default = 1	DECIMAL(10,2)
<b>payment</b>	required	Payment made to the account, debited if < 0	DECIMAL(10,2)
<b>currency</b>	optional	currency of the payment (will be automatically calculated to the user's account currency)	TEXT
<b>vat</b>	optional	Additional VAT %, default=vat of the subuser account	DECIMAL(10,2)
<b>reference</b>	optional	Payment reference, e. g. ID of a subuser	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

#### Notes

- For payments, the type should always be PAYMENT and the V.A.T. should be 0.00.

## Example

The following inserts a payment of 200.00 USD from a creditcard:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=AddAccounting
&subuser=subuser.de
&date=2003-09-03+06:15:00
&description=Credit+Card+Payment
&type=PAYMENT
&vat=0.00
&amount=1
&payment=200.00
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

## 4.5.2 ModifyAccounting

### Description

Mpodify an exisiting accounting entry.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	ID of the subuser account	TEXT
<b>accountingid</b>	required	Accounting ID (as got from command <i>QueryAccountingList</i> )	INTEGER
<b>date</b>	optional	Date of the payment, e. g. 2003-01-01 12:00:00	TEXT
<b>description</b>	optional	Description of this accounting entry	TEXT
<b>type</b>	optional	Type of this accounting entry, e. g. PAYMENT	TEXT
<b>amount</b>	optional	Amount, default = 1	DECIMAL(10,2)
<b>payment</b>	optional	Payment made to the account, debited if < 0	DECIMAL(10,2)
<b>vat</b>	optional	Additional VAT %, default=vat of the subuser account	DECIMAL(10,2)
<b>reference</b>	optional	Payment reference, e. g. ID of a subuser	TEXT
<b>invoiceid</b>	optional	Invoice ID	INTEGER

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

### Notes

- For payments, the type should always be PAYMENT and the V.A.T. should be 0.00

## Example

The following command changes the description of the accounting entry of subuser "subuser.de" with the accountingid 2 to "newdescription":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=ModifyAccounting
&subuser=subuser.de
&accountingid=2
&description="newdescription"
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

### 4.5.3 StatusAccounting

#### Description

This command gives you all available information on a certain accounting.

#### Availability

All users have access to this command.

#### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	ID of the subuser account	TEXT
<b>accountingid</b>	required	ID of the accounting entry	INTEGER

#### Returned properties and values

Property	min	max	Description
ID	0	1	ID of the accounting
DATE	0	1	Date of the accounting
TYPE	0	1	Accounting type, e. g. ADD_DOMAIN
DESCRIPTION	0	1	Description, e. g. a domain name
AMOUNT	0	1	The amount, e. g. the period for domain registrations
PAYMENT	0	1	The total payment of this accounting without V. A. T. in the account currency
VAT	0	1	Additional V. A. T. in percent (%), e. g. 19.00
REFERENCE	0	1	Field for additional information, e.g. subuser who got charged
INVOICEID	1	1	ID of the invoice to which the accounting may be assigned to
CURRENCY	1	1	Currency of the accounting

## 4.5.4 DeleteAccounting

### Description

Delete an accounting entry of a subuser account and update its current account balance.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
subuser	required	ID of the subuser account	TEXT
accountingid	required	ID of the accounting entry	INTEGER

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

### Example

The following deletes the accounting with the ID 3:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command>DeleteAccounting
&accountingid=3
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

## 4.5.5 ExecutePayment

### Description

Execute a payment using an external payment gateway, e.g. to use an interface of a credit card provider or bank debit.

### Availability

All users have access to this command.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>type</b>	required	Payment type, e.g. CREDITCARD	TEXT
<b>amount</b>	required	payment amount	INTEGER
<b>dataN</b>	required	payment data, e.g. for the credit card interface: data0" => "Credit Card Number" data1" => "Credit Card Expiration (YYMM)" data2" => "Credit Card CVC2 code" data4" => "Credit Card Holder" data3" => "Accept AUP:"	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
no properties are returned			

## 4.5.6 QueryAccountingList

### Description

Query the accounting list of the user account or a subuser.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	optional	ID of the subuser account	TEXT
<b>mindate</b>	optional	First accounting date to be returned	DATETIME
<b>maxdate</b>	optional	Last accounting date to be returned	DATETIME
<b>type</b>	optional	Type of accounting	TEXT
<b>description</b>	optional	Description that was entered with the accounting (e.g. domain name)	TEXT
<b>orderby</b>	optional	Sort response by certain criteria: ACCOUNTINGTYPE   ACCOUNTINGDATE   ACCOUNTINGDATEDESC   INVOICEID   REFERENCE	TEXT

## Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
ACCOUNTINGID	0	N	ID of the accounting list entry (only shown if the user is different from the subuser)
ACCOUNTINGDATE	0	N	Date of the accounting list entry
ACCOUNTINGTYPE	0	N	Accounting type, e. g. ADD_DOMAIN
ACCOUNTINGDESCRIPTION	0	N	Description, e. g. a domain name
ACCOUNTINGREFERENCE	0	N	Reference, e. g. ID of the subuser causing that accounting list entry
ACCOUNTINGAMOUNT	0	N	The amount, e. g. the period for domain registrations
ACCOUNTINGPRICE	0	N	The total price of this accounting entry without V. A. T. in the account currency
ACCOUNTINGVAT	0	N	Additional V. A. T. in percent (%), e. g. 16.00
ACCOUNTINGVATPRICE	0	N	Additional V. A. T. in the account currency
OPENINGBALANCEPRICE	1	1	Opening balance without V. A. T.
OPENINGBALANCEVATPRICE	1	1	The V. A. T. included in the opening balance
OPENINGBALANCE	1	1	Opening Balance including V. A. T.
SUMPRICE	1	1	Sum of the opening balance without V. A. T. and the positions without V. A. T.
SUMVATPRICE	1	1	The V. A. T. Included in the sum
SUM	1	1	Sum of the opening balance and the positions

## Notes

- If mindate is without the time part, then “ 00:00:00” will be appended, e. g. “2003-09-01” becomes “2003-09-01 00:00:00”
- If maxdate is without the time part, then “ 23:59:59” will be appended, e. g. “2003-09-01” becomes “2003-09-01 23:59:59”

## Example

The following request queries all accounting list entries with the date “2003-09-01”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=QueryAccountingList
&mindate=2003-09-01
&maxdate=2003-09-01
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[accounting date][0] = 2003-09-01 12:00:03
property[accounting type][0] = ADD_DOMAIN
property[accounting description][0] = test.de
property[accounting reference][0] =
property[accounting amount][0] = 1
property[accounting price][0] = -10.00
property[accounting vat][0] = 16.00
property[accounting vat price][0] = -1.60
property[accounting date][1] = 2003-09-01 14:15:00
property[accounting type][1] = TRANSFER_DOMAIN
property[accounting description][1] = test.com
property[accounting reference][1] = subuser.com
property[accounting amount][1] = 1
property[accounting price][1] = -10.00
property[accounting vat][1] = 16.00
property[accounting vat price][1] = -1.60
property[opening balance price][0] = 100.00
property[opening balance vat price][0] = 0.00
property[opening balance][0] = 100.00
property[sum price][0] = 80.00
property[sum vat price][0] = -3.20
property[sum][0] = 76.80
EOF
```

## 4.6 Miscellaneous commands

The following commands are general commands not related to a specific object class.

### 4.6.1 CheckAuthentication

#### Description

Check the password of a subuser.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>subuser</b>	required	ID of the subuser account	TEXT
<b>password</b>	required	Password of the subuser account	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
530	Authentication failed

Property	min	max	Description
no properties are returned			

#### Example

The following checks the password of "subuser.com":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=CheckAuthentication
&subuser=subuser.com
&password=secret
```

The response is as follows:

```
[RESPONSE]
code = 530
description = Authentication failed
EOF
```

## 4.6.2 CheckAuthorization

### Description

Check if the user is authorized to access an object (checks if the object is within the user's repository).

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
objectclass	required	Class of the object, e. g. DOMAIN	TEXT
objectid	required	ID of the object, e. g. domainname.com	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
531	Authorization failed

Property	min	max	Description
no properties are returned			

### Example

The following checks the authorization for the domain "domainname.com":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=CheckAuthorization
&objectclass=DOMAIN
&objectid=domainname.com
```

The response is as follows:

```
[RESPONSE]
code = 531
description = Authorization failed
EOF
```

The following checks the authorization for the user "subuser.com":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=CheckAuthorization
&objectclass=USER
&objectid=subuser.com
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

### 4.6.3 AssignObject

#### Description

Assign an object to an subuser or to the user itself.

#### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

#### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
s_login	required	Login ID of the user account	TEXT
s_pw	required	Account password	TEXT
s_user	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
command	required	Name of command to be executed	TEXT
objectclass	required	Class of the object. Valid Classes are: DOMAIN, DOMAINTRANSFER, CONTACT, DNSZONE, VSERVER, MOBILEACCOUNT, SSLCERT, DOMAINAPPLICATION	TEXT
objectid	required	ID of the object, e. g. domainname.com	TEXT
user	required	ID of the object's new user	TEXT

#### Returned properties and values

Code	Description
200	Command completed successfully
531	Authorization failed

Property	min	max	Description
no properties are returned			

#### Notes

- The user must have the authorization to access the object as well as the new user

#### Example

The following assigns the domain "domainname.com" to the user "subuser.com":

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=AssignObject
&objectclass=DOMAIN
&objectid=domainname.com
&user=subuser.com
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
EOF
```

#### 4.6.4 SetObjectConfigClass

##### Description

Adds a generic config class to an object. It may be added various text (UTF8) to further describe the object.

##### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

##### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>objectclass</b>	required	Class of the object. Valid Classes are: DOMAIN, DOMAINTRANSFER, CONTACT, DNSZONE, VSERVER, MOBILEACCOUNT, SSLCERT, DOMAINAPPLICATION	TEXT
<b>objectid</b>	required	ID of the object, e. g. domainname.com	TEXT
<b>Configclass</b>	required	Generic description text	TEXT

##### Returned properties and values

Code	Description
200	Command completed successfully
531	Authorization failed

Property	min	max	Description
no properties are returned			

## 4.6.5 GetObjectConfigClass

### Description

Gets the generic config class of an object.

### Availability

All users have access to this command. This command does not make sense if your useraccount does not have any subusers.

### Input parameters

submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>objectclass</b>	required	Class of the object. Valid Classes are: DOMAIN, DOMAINTRANSFER, CONTACT, DNSZONE, VSERVER, MOBILEACCOUNT, SSLCERT, DOMAINAPPLICATION	TEXT
<b>objectid</b>	required	ID of the object, e. g. domainname.com	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
531	Authorization failed

Property	min	max	Description
CONFIGCLASS	1	1	The config class

## 4.6.6 QueryObjectList

### Description

Query a list of objects.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>userdepth</b>	optional	Depth of the list, may be SELF, SUBUSER or ALL	TEXT
<b>first</b>	optional	Index of the first entry to be returned	INTEGER
<b>limit</b>	optional	Max. number of entries to be returned	INTEGER

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
OBJECTCLASS	1	N	The object class
OBJECTID	1	N	The object ID
USER	1	N	The user ID
PARENTUSER	1	N	The user ID of the parent user
COUNT	1	1	The number of entries returned
FIRST	1	1	The index of the first entry
LAST	1	1	The index of the last entry
LIMIT	1	1	The query limit (max. number of entries returned)
TOTAL	1	1	The total number of entries found

### Notes

- userdepth defines the depth of the returned list. SELF returns a list with only objects of the user itself. SUBUSER returns a list with objects of all direct subusers. ALL returns a list with objects of all subusers and all subusers below them.
- first and limit should be used to implement paging

## Example

The following request queries all objects of “reseller.de”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=QueryObjectList
&userdepth=SELF
&first=0
&limit=1000
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[objectclass][0] = DOMAIN
property[objectid][0] = domainname.com
property[user][0] = subuser.de
property[parent user][0] = company
property[objectclass][1] = DOMAIN
property[objectid][1] = mydomain.info
property[user][1] = subuser.de
property[parent user][1] = company
property[objectclass][2] = DOMAINTRANSFER
property[objectid][2] = test.info
property[user][2] = subuser.de
property[parent user][2] = company
property[objectclass][3] = CONTACT
property[objectid][3] = P-ABC123
property[user][3] = subuser.de
property[parent user][3] = company
property[count][0] = 4
property[first][0] = 0
property[last][0] = 3
property[limit][0] = 1000
property[total][0] = 4
EOF
```

## 4.6.7 QueryExchangeRates

### Description

Query list of all current exchange rates. The list represents the daily exchange rates of the ECB (European Central Bank). The rates are used if the object class currency and user account currency are different. See also chapter 2 *UserRelations*.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>mindate</b>	optional	First accounting date to be returned	DATETIME
<b>maxdate</b>	optional	Last accounting date to be returned	DATETIME
<b>first</b>	optional	Index of the first entry to be returned	INTEGER
<b>limit</b>	optional	Max. number of entries to be returned (default = 100)	INTEGER
<b>currencyto</b>	optional	Returns the last exchange rate for the target currency	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
OBJECTCLASS	1	N	The object class
OBJECTID	1	N	The object ID
USER	1	N	The user ID
PARENTUSER	1	N	The user ID of the parent user
COUNT	1	1	The number of entries returned
FIRST	1	1	The index of the first entry
LAST	1	1	The index of the last entry
LIMIT	1	1	The query limit (max. number of entries returned)
TOTAL	1	1	The total number of entries found

### Notes

- userdepth defines the depth of the returned list. SELF returns a list with only objects of the user itself. SUBUSER returns a list with objects of all direct subusers. ALL returns a list with objects of all subusers and all subusers below them.
- first and limit should be used to implement paging

## Example

The following request queries all objects of “reseller.de”:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=QueryExchangeRates
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[date][0] = 2006-06-23
property[currency from][0] = EUR
property[currency to][0] = AUD
property[rate][0] = 1.7096
property[date][1] = 2006-06-23
property[currency from][1] = EUR
property[currency to][1] = BGN
property[rate][1] = 1.9558
property[date][2] = 2006-06-23
property[currency from][2] = EUR
property[currency to][2] = CAD
property[rate][2] = 1.4041
property[date][3] = 2006-06-23
...
property[rate][33] = 1.2502
property[date][34] = 2006-06-23
property[currency from][34] = EUR
property[currency to][34] = ZAR
property[rate][34] = 9.388
property[first][0] = 0
property[last][0] = 34
property[limit][0] = 100
property[count][0] = 35
property[total][0] = 24003
EOF
```

## 4.6.8 QueryObjectRenewalList

### Description

Query a list of objects that have to be renewed. The time interval can be specified with the parameters described in the table below.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>userdepth</b>	optional	Depth of the list, may be SELF, SUBUSER or ALL	TEXT
<b>minexpirationdate</b>	optional	First expiration date to be returned	DATETIME
<b>maxexpirationdate</b>	optional	Last expiration date to be returned	DATETIME
<b>minpaiduntildate</b>	optional	First paiduntildate to be returned	DATETIME
<b>maxpaiduntildate</b>	optional	Last paiduntildate to be returned	DATETIME
<b>minaccountingdate</b>	optional	First accounting date to be returned	DATETIME
<b>maxaccountingdate</b>	optional	Last accounting date to be returned	DATETIME
<b>minfinalizationdate</b>	optional	First finalization date to be returned	DATETIME
<b>maxfinalizationdate</b>	optional	Last finalization date to be returned	DATETIME
<b>minfailuredate</b>	optional	First failure date to be returned	DATETIME
<b>maxfailuredate</b>	optional	Last failure date to be returned	DATETIME
<b>objectid</b>	optional	ID of a object, e. g. "domainname.com"	TEXT
<b>renewalmode</b>	optional	Renewal mode, e. g. "renew"	TEXT
<b>orderby</b>	optional	Order. True values: "OBJECTIDDESC", "OBJECTCLASS", "OBJECTCLASSDESC", "EXPIRATIONDATE", "EXPIRATIONDATEDESC", "PAIDUNTILDATE", "PAIDUNTILDATEDESC", "ACCOUNTINGDATE", "ACCOUNTINGDATEDESC", "FINALIZATIONDATE", "FINALIZATIONDATEDESC", "FAILUREDATE", "FAILUREDATEDESC", "NEXTACTIONDATE", "NEXTACTIONDATEDESC"	TEXT
<b>objectclass</b>	optional	CLASS of the object, e. g. DOMAIN	TEXT
<b>wide = 1</b>	optional	Returns some more properties if set to 1	INTEGER

**Returned properties and values**

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
NEXTACTION	1	N	Next action for the object
FINALIZATIONDATE	1	N	Finalization date
PARENTUSER	1	N	The user ID of the parent user
EXPIRATIONDATE	1	N	Expiration date
ACCOUNTINGDATE	1	N	Accounting date
PAIDUNTILDATE	1	N	Date of the payment
FAILUREDATE	1	N	Failure date
RENEWALMODE	1	N	Renewal mode
OBJECTCLASS	1	N	The object class
OBJECTID	1	N	The object ID
USER	1	N	The user ID
NEXTACTIONDATE	1	N	Date of the next action
FIRST	1	1	The index of the first entry
LAST	1	1	The index of the last entry
LIMIT	1	1	The query limit (max. number of entries returned)
TOTAL	1	1	The total number of entries found

**Notes**

- userdepth defines the depth of the returned list. SELF returns a list with only objects of the user itself. SUBUSER returns a list with objects of all direct subusers. ALL returns a list with objects of all subusers and all subusers below them.
- first and limit should be used to implement paging

## 4.6.9 QueryObjectLogList

### Description

Query a list of objects. For the most objects related options logging entries are created.  
Example: user user.com deletes his domain "reseller.com". His reseller "reseller.com" can see in the "ObjektLogList" that the domain is deleted by user user.com and at which date it is deleted.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>mindate</b>	required	Minimum date of entries	DATE
<b>maxdate</b>	required	Maximum date of entries	DATE
<b>objectid</b>	optional	ID of a object, e. g. "domainname.com"	TEXT
<b>objectclass</b>	optional	CLASS of the object, e. g. DOMAIN	TEXT
<b>orderby</b>	optional	Order. True values: "OBJECTIDDESC", "OBJECTCLASS", "OBJECTCLASSDESC", "LOGDATE", "LOGDATEDESC"	TEXT
<b>operationtype</b>	optional	Search for object log entries with a certain operation type, e.g. CREATE   RENEW   DELETE	TEXT
<b>operationstatus</b>	optional	Search for object log entries with a certain operation status, e.g. SUCCESSFUL   REQUEST   FAILED	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
REFERENCE	1	N	Name of the referring object
LOGDATE	1	N	Date of the logging entry
OBJECTID	1	N	ID of the searched object
OPERATIONTYPE	1	N	Type of the operation
OPERATIONSTATUS	1	N	Status of the operation
LOGINDEX	1	N	ID of the logging entry
LOGUSER	1	N	Name of user who caused the log entry
LOGROLEID	1	N	Name of the role user who caused the log entry
COUNT	1	1	Number of entries

## Example

The following request queries all objects of "reseller.com":

```
BASE-URL?s_login=reseller.com&s_pw=secret
&command=QueryObjectLogList
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[reference][0] = reseller.com
property[logdate][0] = 2005-12-28 11:01:29
property[operationtype][0] = CREATE
property[operationstatus][0] = SUCCESSFUL
property[logindex][0] = 245500
...
EOF
```

## 4.6.10 QueryObjectDeletionLogList

### Description

Query a list of all deleted objects.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret
&command=command
&parameter1=value1
&parameter2=value2
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>objectid</b>	optional	ID of a object, e. g. "domainname.com"	TEXT
<b>objectclass</b>	optional	CLASS of the object, e. g. DOMAIN	TEXT
<b>orderby</b>	optional	Order. True values: OBJECTID   OBJECTIDDESC   OBJECTCLASS   OBJECTCLASSDESC   LOGDATE   LOGDATEDESC	TEXT
<b>mindate/maxdate</b>	optional	min or max date of the log entry	TEXT

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
REFERENCE	1	N	Name of the referring object
LOGDATE	1	N	Date of the logging entry
LOGINDEX	1	N	ID of the logging entry
OBJECTCLASS	1	N	Class of the object
OBJECTID	1	N	ID of the object
LOGUSER	1	N	Name of user who caused the log entry
LOGROLEID	1	N	Name of the role user who caused the log entry
COUNT	1	1	Number of entries

## 4.6.11 StatusObjectLog

### Description

Gives you detailed information on one object listed in the object log.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>logindex</b>	required	ID of the logging entry	INTEGER

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
OBJECTCLASS	1	1	Object class
LOGDATE	1	1	Date of the logging entry
LOGSTATUS	1	1	Status of the logging entry, e. g. "ok"
OBJECTID	1	1	Object ID
OPERATIONTYPE	1	1	Type of the operation
OPERATIONSTATUS	1	1	Status of the operation
OPERATIONSINFO	1	N	The complete command and the response as text. Each array entry represents a line.
LOGUSER	1	N	Name of user who caused the log entry
LOGROLEID	1	N	Name of the role user who caused the log entry
REFERENCE	1	1	Reference of the user who caused the log entry
NEWOBJECTPROPERTIES	1	N	The new properties of an object after e.g. an update

## Example

The following request queries the objects log with the logindex "754297":

```
BASE-URL?s_login=reseller.com&s_pw=secret
&command=StatusObjectLog
&logindex=754297
```

The response is as follows:

```
[RESPONSE]
code = 200
description = Command completed successfully
property[logdate][0] = 2006-06-05 14:19:35
property[logstatus][0] = ok
property[objectclass][0] = DNSZONE
property[objectid][0] = reseller.com.
property[operationtype][0] = UPDATE
property[operationstatus][0] = REQUEST
property[operationinfo][0] = [COMMAND]
property[operationinfo][1] = DELRR0=www.reseller.com. 3600 IN A 192.168.0.1
property[operationinfo][2] = DNSZONE=reseller.com.
property[operationinfo][3] = COMMAND=UpdateDNSZone
property[operationinfo][4] = ADDR0=www.reseller.com. 3600 IN A 192.168.0.2
property[operationinfo][5] = INCSERIAL=1
property[operationinfo][6] = EOF
property[operationinfo][7] =
property[operationinfo][8] = [RESPONSE]
property[operationinfo][9] = code = 200
property[operationinfo][10] = description = Command completed successfully
property[operationinfo][11] =
property[operationinfo][12] =
property[operationinfo][13] =
property[operationinfo][14] = EOF
...
EOF
```

## 4.6.12 StatusObjectDeletionLog

### Description

Gives you detailed information on one object listed in the object deletion log.

### Availability

All users have access to this command.

### Input parameters

Submit the request using the following syntax:

```
BASE-URL?s_login=reseller.de&s_pw=secret  
&command=command  
&parameter1=value1  
&parameter2=value2  
&parameter3=value3...
```

Parameter	Obligation	Definition	Type
<b>s_login</b>	required	Login ID of the user account	TEXT
<b>s_pw</b>	required	Account password	TEXT
<b>s_user</b>	optional	User ID of the user as which the command shall be executed (must be a subuser of s_login)	TEXT
<b>command</b>	required	Name of command to be executed	TEXT
<b>logindex</b>	required	ID of the logging entry	INTEGER

### Returned properties and values

Code	Description
200	Command completed successfully
541	The command failed

Property	min	max	Description
OBJECTCLASS	1	1	Object class
LOGDATE	1	1	Date of the logging entry
OBJECTID	1	1	Object ID
LOGUSER	1	N	Name of user who caused the log entry
LOGROLEID	1	N	Name of the role user who caused the log entry
REFERENCE	1	1	Reference of the user who caused the log entry
FINALPROPERTY	1	N	The final properties of an object before it got deleted

## 5. Additional Services

The following services are add-on services for the Core API.

### 5.1 POP3-Event-Server

The POP3-Event-Server can be used to receive event by the POP3 protocol (specified by RFC 1939).

The following configuration data is required to configure the POP3-Client:

Server	194.50.187.138
Login	<User ID used with the API>
Password	<Password of the User>

The following example shows an email that is generated by the POP3-Event-Server:

```
From: EVENT MANAGER <events@invalid.example>
To: USER <user@invalid.example>
Date: Mon, 10 Apr 2006 15:39:22 -0000
Subject: [DOMAINAPPLICATION::FAILED] 44361429E6E0F47329937
```

```
EVENT_DATE:      2006-04-10 15:39:22.0
EVENT_CLASS:     DOMAINAPPLICATION
EVENT_SUBCLASS:  FAILED
OBJECT_CLASS:    APPLICATION
OBJECT_ID:       44361429E6E0F47329937
```

```
[DATA]
application:44361429E6E0F47329937
domain:reseller.com
reason:HOLD
```

## 6. System response code list

Code	Description
200	Command completed successfully
210	Domain name available
211	Domain name not available
212	Name server available
213	Name server not available
214	Contact available
215	Contact not available
216	Object available
217	Object not available
220	Command completed successfully. Server closing con...
420	Command failed due to server error. Server closing...
421	Command failed due to server error. Client should ...
425	Service temporarily locked; usage exceeded
500	Invalid command name
501	Invalid command option
502	Invalid entity value
503	Invalid attribute name
504	Missing required attribute
505	Invalid attribute value syntax
506	Invalid option value
507	Invalid command format
508	Missing required entity
509	Missing command option
510	Command not supported for this class
520	Server closing connection. Client should try openi...
521	Too many sessions open. Server closing connection
530	Authentication failed
531	Authorization failed
532	Domain names linked with name server
533	Domain name has active name servers
534	Object has not been flagged for transfer
535	Restricted IP address
536	Object already flagged for transfer
540	Attribute value is not unique
541	Invalid attribute value
542	Invalid old value for an attribute
543	Final or implicit attribute cannot be updated
544	Entity on hold
545	Object not found
546	Credit limit exceeded
547	Invalid command sequence
548	Object is not up for renewal
549	Command failed
550	Parent object not registered

Code	Description
551	Parent object status does not allow for operation
552	Object status does not allow for operation
553	Operation not allowed. Object pending transfer
554	Object already registered
555	Object already renewed
556	Maximum registration period exceeded

## 7. ChangeLog

Date	Old / new version	Description of changes
2007-07-06	1.5 / 1.6	Added description of new price relations TRANSFERAPPLICATION and RENEWPERIOD
2007-12-24	1.6 / 1.7	Added commands GetObjectConfigClass and SetObjectConfigClass
2008-01-29	1.7 / 1.8	Added command section "User role management commands (ACL)"
2008-02-20	1.8 / 1.9	Added system response code list
2008-02-20	1.9 / 1.9	Added parameter "userdepth" to command QueryRoleUserList
2008-03-05	1.9 / 1.9	Added parameter "currency" to command AddAccounting
2008-05-08	1.9 / 2.0	Added parameters "wide" and "userN/userdepthN" to command QueryUserList
2008-05-28	2.0 / 2.1	Added parameters to QueryObjectRenewalList
2008-06-10	2.1 / 2.1	Added parameter "matchmode" to command FindUser
2008-06-13	2.1 / 2.1	Added standard parameters first and limit to QueryUser
2008-07-03	2.1 / 2.2	Added parameter "stats" to command StatusUser
2008-07-08	2.2 / 2.2	Added parameter "currencyto" to command QueryExchangeRates
2008-07-24	2.2 / 2.3	Added several parameters in several commands
2008-10-30	2.3 / 2.4	Added some parameters to command "QueryObjectLogList" Added command QueryObjectDeletionLogList
2008-11-17	2.4 / 2.4	Changed response parameters in command "FindUser"
2008-11-21	2.4 / 2.4	Added parameter "orderby" to command "QueryAccountingList"
2009-01-13	2.4 / 2.5	Added several parameters so several commands ; Added command "StatusObjectDeletionLog"
2009-01-26	2.5 / 2.5	Added command "StatusAccounting"
2009-01-29	2.5 / 2.6	Added several parameters to command "QueryUserList"
2009-02-10	2.6 / 2.6	Added parameter "orderby" to command "QueryUserList"