

XML DOCUMENTATION

XML Interface Documentation

Version 16.0

June 2016

© 1998-2016

Text, contents, design and the arrangement of this manual are subject to the protection of copyright laws. You may only use or copy its content for private use, personal or non-commercial purposes and while respecting the provisions of copyright laws.

New Feature Polling

Please contact our domain support for activating the polling feature.

Table of Contents

1 The XML Interface	1
1.1 Introduction.....	1
1.2 General Tasks.....	1
1.3 General Keys.....	10
1.4 Implementations of the Interface	13
2 Domain Processes	17
2.1 Synchronous and Asynchronous Replies.....	17
2.2 Domain and Domain Contact Transactions.....	18
2.3 Transfer-In Transactions and Owner Change	19
2.4 Transfer-Outs	21
2.5 Zone, User and Inquire Transactions	23
3 Polling	24
3.1 General Information.....	24
3.2 Messages	27
3.3 Domain Jobs	28
3.4 Domain Transfer Jobs	36
3.5 Contact Jobs.....	45
4 Domain.....	50
4.1 Description of the XML Keywords.....	50
4.2 Domain Create (0101).....	54
4.3 Domain Create Plus (0101).....	55
4.4 Domain Update (0102).....	59
4.5 Domain Update Plus (0102)	61
4.6 Domain Inquire Tasks (0105)	66
4.7 Domain Delete (0103)	72
4.8 Domain Ownerchange (0104010)	73
4.9 Domain Renew (0101003)	75
4.10 Domain Premium Inquire (0165)	76
4.11 Domain Premium Info (0164)	77

4.12 Domain Premium Info (0164)	78
4.13 Domain Premium List (0165)	79
5 Preregistration.....	84
5.1 Description of the XML Keywords	84
5.2 Domain Prereg Create (0110001).....	86
5.3 Domain Prereg Create Plus (0110001)	87
5.4 Domain Prereg Confirm (0110007)	89
5.5 Domain Prereg Update (0110002)	90
5.6 Domain Prereg Delete (0110003).....	91
5.7 Domain Prereg Inquire Tasks (0110004, 0110005).....	92
5.8 Domain Prereg Info (0110004)	93
5.9 Domain Prereg List (0110005)	94
5.10 Prereg Config List (0110006).....	96
6 Domain Cancellation and Domain Restore	98
6.1 Description of the XML Keywords	98
6.2 Domain Cancellation Create (0103101).....	99
6.3 Domain Cancellation Update (0103102)	100
6.4 Domain Cancellation Delete (0103103).....	101
6.5 Domain Cancellation Inquire Tasks (0103104).....	102
6.6 Domain Cancellation Info (0103104)	103
6.7 Domain Cancellation List (0103104)	104
6.8 Domain Cancellation History Inquire Tasks (0103124, 0103125).....	105
6.9 Domain Cancellation History Info (0103124).....	106
6.10 Domain Cancellation History List (0103125)	107
6.11 Domain Restore (0101005)	109
6.12 Domain Restore List (0105005)	110
7 Domain Transfer.....	114
7.1 Description of the XML Keywords	114
7.2 Domain Transfer In (0104)	118
7.3 Domain Transfer Plus (0104)	120
7.4 Domain Transfer Plus with Ownerchange (0104010)	124
7.5 Domain Transfer-Out (0106002)	127

7.6 Transfer-Out Inquire Tasks (0106001).....	128
7.7 TransferOut Info (0106001).....	129
7.8 TransferOut List (0106001).....	130
7.9 Domain Status (0102002)	131
7.10 AuthInfo1Create (Apply for AuthInfo) (0113001).....	133
7.11 AuthInfo Delete (0113002)	135
7.12 AuthInfo2 Create (Apply for AuthInfo2) (0113003).....	136
7.13 IRTP Domain Info (0104008)	138
7.14 Irtp DomainRestart (114002)	141
8 Domain Backorder	142
8.1 Backorder Create (0141).....	142
8.2 Backorder Delete (0143)	143
8.3 Domain Backorder User List.....	144
8.4 Domain Backorder List (0146)	146
9 Zone.....	149
9.1 Description of the XML Keywords.....	149
9.2 Zone Create (0201)	153
9.3 Zone Update (0202, 0202001).....	154
9.4 Zone Import (0204).....	155
9.5 Zone Delete (0203)	155
9.6 Zone Inquire Tasks (0205)	156
9.7 Zone Info (0205).....	158
9.8 Zone List (0205).....	159
10 DNSSEC.....	161
10.1 Description of the XML Keywords.....	161
10.2 DNSSec Sign (0120001).....	162
10.3 DNSSec Key-Rollover (0120002)	162
10.4 DNSSec Unsign (0120003)	164
10.5 DNSSec Inquire Tasks (0120004, 0120005)	165
10.6 DNSSec Info (0120004)	166
10.7 DNSSec List (0120005)	167

11 BackupMX	168
11.1 Description of the XML Keywords.....	168
11.2 BackupMX Create (0521)	168
11.3 BackupMX Inquire (0524)	169
11.4 BackupMX Info (0524).....	170
11.5 BackupMX List (0524).....	171
11.6 BackupMX Delete (0523)	172
12 MailProxy	173
12.1 Description of the XML Keywords.....	173
12.2 Mailproxy Create (0511)	177
12.3 MailProxy Update (0512)	178
12.4 MailProxy Inquire (0514)	180
12.5 Mailproxy Info (0514).....	181
12.6 MailProxy Delete (0513).....	183
13 Domain Contacts (Handles)	185
13.1 Description of the XML Keywords.....	185
13.2 Contact Create (0301)	187
13.3 Contact Update (0302).....	189
13.4 Contact Comment Update (0302004).....	191
13.5 Contact Delete (0303)	191
13.6 Contact Inquire Tasks (0304)	192
13.7 Contact Info (0304).....	193
13.8 Contact List (0304).....	195
13.9 Additional Information for Domain Contacts (Extensions)	196
13.10 Extensions for .aero Domains	197
13.11 Extensions for .au domains (Australia)	197
13.12 Extensions for .barcelona domains	199
13.13 Extensions for .cat Domains (Catalan language area)	199
13.14 Extensions for .hk Domains	199
13.15 Extensions for .it Domains	200
13.16 Extensions für .jobs Domains.....	200
13.17 Extensions for .ro Domains	201

13.18 Extensions for .ru Domains (Russia).....	202
13.19 Extensions for .swiss domains	202
13.20 Extensions for .uk Domains	202
13.21 Extensions for .xxx Domains	203
14 Domain Contact Verification	204
14.1 Description of the XML Keywords.....	204
14.2 Contact Verification Create (0341)	205
14.3 Contact Verification Inquire Tasks (0344, 0345).....	205
14.4 Contact Verification Info (0344).....	206
14.5 Contact Verification List (0345).....	208
14.6 Contact Verification Resend Email (0349).....	209
15 Redirects.....	210
15.1 Description of the XML Keywords.....	210
15.2 Redirect Create (0501)	211
15.3 Redirect Update (0502).....	213
15.4 Redirect Delete (0503).....	215
15.5 Redirect Inquire Tasks (0504)	216
15.6 Redirect Info (0504)	217
15.7 Redirect List (0504).....	218
16 User	220
16.1 Description of the XML Keywords.....	220
16.2 User Create (1301001)	223
16.3 User Update (1301002).....	225
16.4 User Inquire Tasks (1301004)	226
16.5 User Info (0110004).....	227
16.6 User List (1301004).....	228
16.7 User Profile Update (1301014)	229
16.8 User Copy (1301016).....	230
16.9 User Delete (1301003)	231
16.10 Object User Assignment (1308).....	232
16.11 Object User Assignment All (1308001).....	233
16.12 Object User Assignment Bulk (1308002)	233

17 Inquiring Requests	235
17.1 Description of XML Keywords.....	235
17.2 History Inquire Tasks (0713)	236
18 Addendum	239
18.1 Changelog.....	239
18.2 Formats	242
18.3 CodeList	243
18.4 User Statuses	245
18.5 .HK: Industry Types	246

1 The XML Interface

1.1 Introduction

The backend communicates with XML templates. Every XML request is validated to check if the login credentials match and if the request contains valid syntax before it is parsed. If successful, the system then sends a XML response containing the result of the transaction to the user. The response displays the affected data and a result status respectively.

Depending on the type of order, the system will switch between using the database data for real time communication or via communication with the registry for asynchronous processing.

The following task types exist:

- Info task (Single Inquire): a single query for an object such as a certificate or contact
- List task (List Inquire) a query for a group of similar objects (example: multiple certificates)
- Create task: creation of new objects in the database (e.g. a contact) as well as at the registry (e.g. a certificate)
- Update task: changing an existing object - example a contact
- Delete task: permanently removing an existing object - example a contact

1.2 General Tasks

1.2.1 Default Request

Each request sent to the domain management system requires an *auth* element, containing the tags *user*, *password* and *context*.

NOTE

The system only works with UTF-8.

NOTE

When doing an UTF-8 conversion with Microsoft.NET, keep in mind that the UTF-8 encoded XML String may not contain a Byte Order Mark (BOM ï»¿). A BOM is set by default at the beginning of the string.

Key	Occurs	Format	Description
request	1		Request
request / auth	1		Auth-Block
request / auth / user	1	string	User name
request / auth / password	1	string	User password
request / auth / context	1	string	ID of the personalized system of the subuser.
request / owner	0 - 1		Subuser
request / owner / user	1	string	Username of the subuser
request / owner / context	1	string	ID of the personalized system of the subuser.
request / language	0 - 1	enum	Overwrites the language of error messages. <ul style="list-style-type: none"> • en • de
request / task	1		The task. Several task blocks can be used.
request / task / code	1	string	The task code

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <owner>
    <user>customer2</user>
    <context></context>
  </owner>
  <language>de</language>
  <task>
    <code>0130</code>
  </task>
</request>
```

1.2.2 Default Response

Key	Occurs	Format	Description
response	1		Response of the system
response / result	1 - n		The result block. For multitasks several result blocks will be sent.
response / result / data	0 - n		Includes the returned data.
response / result / data / object	1-n	string	Includes the data objects.
response / result / status	1		

Key	Occurs	Format	Description
response / result / status / type	1	string	The status type of the response <ul style="list-style-type: none"> • success (S) • error (E) • notification (N)
response / result / status / code	1	string	The response code (N, E, S)
response / result / status / text	1	string	Details concerning the status
response / result / msg	0 - n		The system message
response / result / msg / code	1	string	The code of the task, the message belongs to. E.g S0101: S = Succes, 0101= Domain Create
response / result / msg / text	1	string	The text of the message
response / result / msg / object	0 - n	string	The object the message refers to.
response / result / msg / object / type	0 - n	string	The object type, e.g. "domain" or "contact." The object type, e.g. "certificate" or "contact."
response / result / msg / object / value	0 - n	string	The object value, e.g. z. B. "example.com" (type = domain) The objectvalue, e.g. z. B. "12345" (type = certificate ID)
response / result / msg / help	0 - n	string	Notes

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <object></object>
      <object></object>
    </data>
    <status>
      <type></type>
      <code></code>
      <text></text>
    </status>
    <msg>
      <code></code>
      <type></type>
      <text></text>
      <help></help>
      <object>
        <type></type>
        <value></value>
      </object>
      <object>
        <type></type>
        <value></value>
      </object>
    </msg>
    <msg>
      <code></code>
      <type></type>
      <text></text>
      <object>
        <type></type>
        <value></value>
      </object>
    </msg>
  </result>
</response>
```

```

        </object>
        <object>
            <type></type>
            <value></value>
        </object>
        <help></help>
    </msg>
</result>
</response>

```

1.2.3 Status Codes and their Meanings

Syntax	Example	Meaning
"S" and task code	S0101	Success = request successfully accepted
"E" and task code	E0101	Error = an error occurred
"N" and task code	N0101	Notification = for some requests you first receive a „Notify“ and not a „Success“ or „Error“ message. About the final status you will be informed via email. To be able to assign the messages to the object, we recommend you to use a unique transaction id (ctid).
"EF" and task code	EF00001	EF (Error Function) = notice of a processing error supplemented by a specific error code

In this example we inform you about the acknowledgement of a request. After the acknowledgement you receive a message if the request was successful or not.

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0101</code>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
      <text>Domain registration successfully started</text>
      <text>.</text>
      <type>notify</type>
    </status>
  </result>
</response>

```

1.2.4 Default Inquiries

a.) Single Inquiries

With a single inquire you ask for the detailed data of a single object. With the `key` key additional data can be inquired. See the respective chapter for details.

Key	Occurs	Format	Description
code	1	task	E.g. Domain Inquire 0105
object type	1		The required key depends on the type of the inquired object. E.g. "domain", "contact" or "zone".
object type / identifier	1	string	E.g. name id , the name of the inquired object.
key	0 - n		Key to inquire additional values. The available values depend on the object type.

Example (domain inquire)

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0105</code>
    <view>
      <offset>0</offset>
      <limit>30</limit>
      <children>1</children>
    </view>
    <where>
      <and>
        <key>name</key>
        <operator>eq</operator>
        <value>test*</value>
      </and>
      <and>
        <or>
          <key>name</key>
          <operator>eq</operator>
          <value>*.de</value>
        </or>
        <or>
          <key>name</key>
          <operator>eq</operator>
          <value>*.com</value>
        </or>
      </and>
    </where>
  </task>
</request>
```

b.) List Inquiries

With a list inquire you ask for the data of several or all objects of an object type. The list inquiries show less details about each object than a single inquire. With the `key` key you can inquire additional object data. See the respective chapters for details.

Note

To run a list inquire of `sld`, `tld`, `subtld` include them into a "where" bloc. The system response displays the complete domain names not only the key data, e.g. ".com".

Key	Occurs	Format	Description
view	1		Contains the objects to be inquired.
view / offset	0 - 1	string	The position of the object the inquiry should start with.
view / limit	0 - 1	string	The maximum number of objects to be inquired.
view / children	0 - 1	enum	Inquire subusers: <ul style="list-style-type: none"> • 1 = inquire subusers too • 0 = do not inquire subusers
where	0 - 1	string	The <code><where></code> element contains the description of the object to be queried. In the <code><where></code> element you can interleave multiple search terms by using the keys <code><where>/*/<and></code> and <code><where>/*/<or></code> . The key <code><and></code> always has precedence over <code><or></code> .
where / key	0 - 1	string	Type of the key
where / operator	0 - 1	string	Operator All possible types of operators are listed below.
where / value	0 - 1	string	Key value
key	0 - 1	string	Inquiring the values of a special key; possible keys depend on the task, e.g.: <ul style="list-style-type: none"> • owner • name • id • created
order			Sorting

Key	Occurs	Format	Description
order / key	1	string	Keyword by which the data is to be sorted
order / mode	0 - 1	enum	Sort sequence asc = ascending desc = descending Default = asc

In the following example the user asks for his own domains and the domains of his sub users (children =1). The list contains the first thirty domains (offset = 0, limit = 30) beginning with "a" (name like a*). Additionally the key created is inquired (key = created).

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0105</code>
    <view>
      <offset>0</offset>
      <limit>30</limit>
      <children>1</children>
    </view>
    <key>created</key>
    <where>
      <key>name</key>
      <operator>eq</operator>
      <value>a*</value>
    </where>
    <order>
      <key>created</key>
      <mode>asc</mode>
    </order>
  </task>
</request>
```

Example for a "where" Condition in a Domain Request

This request asks for all domains beginning with "test" and belonging to the TLDs "de" or "com".

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0105</code>
    <view>
      <offset>0</offset>
      <limit>30</limit>
```

```

        <children>1</children>
    </view>
    <where>
        <and>
            <key>name</key>
            <operator>eq</operator>
            <value>test*</value>
        </and>
        <and>
            <or>
                <key>name</key>
                <operator>eq</operator>
                <value>*.de</value>
            </or>
            <or>
                <key>name</key>
                <operator>eq</operator>
                <value>*.com</value>
            </or>
        </and>
    </where>
</task>
</request>

```

Following Operators are allowed:

Value	Meaning
eq (equal)	equal to
ne (not equal)	not equal to
like	wildcard search "*"
gt (greater then)	greater than
lt (lower then)	lower than
ge (greater or equal)	greater than or equal to
le (lower or equal)	less than or equal to

1.2.5 Multitasks

Multitasks are used to send several tasks in a single request. Only one auth block is necessary.

Example

```

<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>9</context>
  </auth>
  <task>
    <code>0130</code>
    <ctid>100001</ctid>
  </task>

```



```

<task>
  <code>0101</code>
  <ctid>100002</ctid>
</task>
</request>

```

1.2.6 Bulk Request

With a bulk request several objects can be created, updated or deleted with a single task.

NOTE

A bulk request is not possible for all functions.

Example (bulk request, domain create)

```

<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0101</code>
    <ctid></ctid>
    <default>
      <ownerc>9925345</ownerc>
      <adminc>9925345</adminc>
      <techc>9925345</techc>
      <zonec>9925345</zonec>
      <nserver>
        <name>ns1.example.org</name>
        <ip>172.16.100.1</ip>
      </nserver>
      <nserver>
        <name></name>
        <ip></ip>
      </nserver>
      <dns_checks>soa</dns_checks>
    </default>
    <domain>
      <name>example.org</name>
    </domain>
    <domain>
      <name>example.net</name>
    </domain>
    <domain>
      <name>example.de.de</name>
      <ownerc>6574839</ownerc>
      <!--
        Here an ownerc deviant from the default value is used.-->
    </domain>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>

```

1.3 General Keys

Key	Occurs	Format	Description
request / owner	0 - 1		With the owner block you can carry out requests for a subuser.
request / owner / user	1	string	Username of the subuser
request / owner / context	1	string	ID of the personalized system of the subuser
request / task / reply_to	0 - 1	string	Contact address for our support team in case of errors. Required for domain and domain contact requests. Not required for the other requests.
request / task / email_confirmation	0 - 1	string	The successful receipt of the request will be confirmed via email.
request / task / execdate	0 - 1	execdate	Time of execution. Format:yyyy-MM-dd
			NOTE Please note, that this key is not valid for all tasks.
request / task / ctid	0 - 1	string	= Client Transaction ID. Ctid allows you to automate procedures, because email replies always include the CTID. In a multitask you can find it in each result tag.
request / bulk_limit	0 - 1	int	Bulk request means that tasks are received, forwarded in a bulk queue and processed from there. The system answer contains one "result" block for each request. Large bulk requests cause long system message which slow down processing. For this reason the bulk_limit key is useful. If a bulk request contains more than 20 requests the system only sends a short confirmation message. All further details are sent via email.

1.3.1 Owner

Format

```
...
<owner>
  <user>customer</user>
  <context></context>
</owner>
<task>
  <code></code>
  <object></object>
</task>
```

Example

Registering a domain for a subuser, which will be assigned to this user.

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
```

```

    <password>password</password>
    <context>1049</context>
  </auth>
  <owner>
    <user>customer2</user>
    <context></context>
  </owner>
  <task>
    <code>0101</code>
    <domain>
      <name>example.com</name>
      <ownerc>9925346</ownerc>
      <adminc>9925346</adminc>
      <techc>9925346</techc>
      <zonec>9925346</zonec>
      <nserver>ns1.example.com</nserver>
    </domain>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <owner>
    <user>customer2</user>
    <context></context>
  </owner>
  <task>
    <code>400101</code>
    <certificate>
      <admin>
        <id>14</id>
      </admin>
      <technical>
        <id>14</id>
      </technical>
      <name>example.com</name>
      <approver_email>admin@example.com</approver_email>
      <product>SSL123</product>
      <lifetime>12</lifetime>
      <expire>2015-05-11 23:49:00</expire>
      <software>APACHESSL</software>
      <csr>
        -----BEGIN CERTIFICATE REQUEST-----
        PIIC4zCCAcCAQAwgZ0xCzAJBgNVBAYTAkRFMRMwEQYDVQQIDApPcGVycG
        ZlbHR6MRUwEwYDVQQHDAxSZWdlbnNjaGJ1cmcxEjaQBGNVBAoMCUudGVy
        bmV0WDEUMBIGAlUECwwLRW50d2lja2x1bmcxFzAVBgNVBAMMDnd3dy50ZX
        NOMTIzLmRlMR8wHQYJKoZIhvcNAQkBFhBhZGlpbkBoZ0ZlZGZlZGZlZGZl
        jANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxqO+nSNNzA71zVvY0kcv
        JULQUWFkRXLWjLcVhHOidqyX1znC5E9n9JepxEhxU4gVjSc2+2IfmIMtzMa
        9Sv+JT0xJb6TFsChKx6Xq2S9UMzaywQdnk6UOv8nr00Tz4YT1jG2Zj2HN/
        erEsJ61ReQTWiB7cgerJQolv9PSgG1GMHe9wRJ57m0aob90T5'udoj6Df4Mi
        LgEuXJzITH5XHwJdWsDvkmAGrARasxcAVRdfZXE2eQHQP5+8tOf0oiHncqzW
        Cr+qA+oEhfD2MCDTyZLXvguFnqxLDxvTweFALaNCp5n930ok3IZPACV4rMoQ
        Z5o8k/GM/QJmUtl1lapuyMhfwIDAQABoAAwDQYJKoZIhvcNAQEFBQADggEBA
        L1E2RR2FC85pf7y2qLbAJ8GjTHW5tLK6j0YqfFQEvicBdc+aPoguo/0xWg6U
        T4VYJVBHYZ1zdfeED5JfRs5J1TRo9TzEr67tZ00We2n+NaOitKclnPfAjt/u
        GccZmL6zzmlVaJDMf1b8zxEw+aejq/KqJOGwLwFYsq7HeGiZG+KNmf7Rtvu+T
        ZGU326+k0GL1i5jRdDUEkrrg/56rzKtVC5gs9ndsSZ6UNYePe2x9sHCxzHrL5
        OgsFTnddKFMf7X+tbRZF2QKYHQ1TtHnhIClNRis8foiTFj7wP8z8desiWYAi
        jWXwHEMIQ4UuP8KuB2M6MO+fM+gd6K8HvPkCk6C9dQ=
        -----END CERTIFICATE REQUEST-----
      </csr>
    </certificate>
    <subject_alternative_name>

```

```

        <name>www.example.com</name>
        <approver_email>customer@example.com</approver_email>
        <order_id></order_id>
    </subject_alternative_name>
    <subject_alternative_name>
        <name>www.example1.com</name>
        <approver_email>customer@example.com</approver_email>
        <order_id></order_id>
    </subject_alternative_name>
    <subject_alternative_name>
        <name>www.example2.com</name>
        <approver_email>customer@example.com</approver_email>
        <order_id></order_id>
    </subject_alternative_name>
</certificate>
</task>
</request>

```

1.3.2 Reply to

Example

```

...
<task>
...
    <reply_to>customer@example.com</reply_to>
...
</task>
...

```

1.3.3 Email Confirmation

Example

```

...
<task>
...
    <email_confirmation>customer@example.com</email_confirmation>
...
</task>
...

```

1.3.4 Ctid (= Client Transaction ID)

Example

```

...
<task>
...
    <ctid>123456</ctid>
...
</task>

```

NOTE

Use a unique Ctid for each request, to be able to assign each system reply to the right request. Use only the characters a-z, A-Z, 0-9. The Ctid should be one string without blank spaces.

1.3.5 Bulk_limit

Example

```
<request>
...
<bulk_limit>100</bulk_limit>
<task>
...
</task>
</request>
```

1.4 Implementations of the Interface

You can access the HTTPS interface of the system via <https://gateway.autodns.com>.

1.4.1 Perl Client (LWP)

```
#!/usr/local/bin/perl

use LWP::UserAgent;

$ua = new LWP::UserAgent;
$ua->agent('AgentName/0.1 ' . $ua->agent);
$ua->timeout(1200);

my $req = new HTTP::Request POST => ;

$req->content_type('text');
my $content = getFileContent('request.xml');
$req->content($content);

my $time = localtime();
my $res = $ua->request($req);

print "HTTP Request ($time):\n".$req->as_string()."\n";

if ($res->is_success) {
$res_str = "HTTP Antwort:\n".$res->as_string()."\n";
open( Antwort, ">$ARGV[1]" );
```

```
print Antwort $res->as_string();
close( Antwort );
$res_str=~s/^\<request>\n.*\n</request>$//sm;
print $res_str;
} else {
print($res->error_as_HTML."\n");
}

sub getFileContent {
my ($FileHandle , $FileContent, $FileLength);
open(FileHandle, "<$_[0]") or die "Can't open file '$_[0]': $!";;
$FileLength = sysseek(FileHandle, 0, 2 );
if( $FileLength<=0 ) { die "file \"$_[0]\" is empty!"; }
sysseek(FileHandle, 0, 0 );
sysread(FileHandle,$FileContent,$FileLength);
close(FileHandle);
return $FileContent;
}
```

1.4.2 PHP-Client (for PHP 4 and PHP 5 via cURL)

```
<?php
define( 'HOST', );
define( 'XML_FILE', 'request.xml' );
$xml = implode( "", file(XML_FILE) );
header( 'Content-Type: text/xml' );
echo requestCurl( $xml );

function requestCurl( $data ) {
$ch = curl_init( HOST );
curl_setopt ( $ch, CURLOPT_POSTFIELDS, $data );
curl_setopt ( $ch, CURLOPT_RETURNTRANSFER, 1 );
curl_setopt ( $ch, CURLOPT_SSL_VERIFYPEER, FALSE );
if( !$data = curl_exec( $ch ) ) {
echo 'Curl execution error.', curl_error( $ch ) ."\n";
return FALSE;
}
curl_close( $ch );
return $data;
}
?>
```

1.4.3 Java Client

```
import java.io.BufferedReader;
import java.io.DataOutputStream;
```

```
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.Charset;

/**
 * Communicates with the backend.
 *
 * @author
 *
 */
public class Communicator {

    private static final String BACKEND_URL = "https://gateway.autodns.com/";

    private static final Charset utf8Charset = Charset.forName("UTF-8");

    /**
     * sends the given request to the predefined backend url via http post and reads the
     response
     *
     * @param request
     * @return
     */
    public String sendRequest(String request) {
        URL url;
        HttpURLConnection connection = null;
        try {

            byte[] requestBytes = request.getBytes(utf8Charset);

            // Create connection
            url = new URL(BACKEND_URL);
            connection = (HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setRequestProperty("Content-Type", "application/x-www-form-
            urlencoded");
            connection.setRequestProperty("HTTP-Version", "HTTP/1.0");
            connection.setRequestProperty("Content-Length", "" +
            Integer.toString(requestBytes.length));
            connection.setRequestProperty("Content-Language", "de-DE");
            connection.setRequestProperty("charset", "utf-8");
            connection.setUseCaches(false);
            connection.setDoInput(true);
            connection.setDoOutput(true);
```

```
// Send request
DataOutputStream dataOutputStream = new
DataOutputStream(connection.getOutputStream());
dataOutputStream.write(requestBytes);
dataOutputStream.flush();

// Get Response
InputStream is = connection.getInputStream();
BufferedReader reader = new BufferedReader(new InputStreamReader(is));
String line;
StringBuffer response = new StringBuffer();
while((line = reader.readLine()) != null) {
    response.append(line);
    response.append("\r");
}
return response.toString();
} catch(Exception e) {
    e.printStackTrace();
    return null;
} finally {
    if(connection != null) {
        try {
            connection.getOutputStream().close();
            connection.getInputStream().close();
        } catch(Throwable e) {
        }
        connection.disconnect();
    }
}
}
```

NOTE

Alternatively you can use the following client:

<http://hc.apache.org/httpcomponents-client-ga/examples.html>

2 Domain Processes

2.1 Synchronous and Asynchronous Replies

When sending requests to the domain management system, you can receive synchronous (real time) or asynchronous replies. The types S, E and N show you the status of your request.

ReplyType	Meaning	Example
S (Success)	The request was successfully processed.	<pre><system> user: sample_user task: 0101 status: S0101[The domain has been successfully registered.] </system></pre>
E (Error)	An error occurred and the request was not completely processed.	<pre><system> user: sample_user task: 0103 (delete domain) status: E0103 [Domain could not be deleted.] </system></pre>
N (Notification)	The request has been accepted. After being processed completely, a asynchronous reply will be sent.	<pre><system> user: sample_user task: 0101 status: N0101] [Domain-registration was successfully started.] </system></pre>

Synchronous reply:

After receiving your request, will send you a reply with the status of your request. The reply returns in the same way that you have sent the request to the system. That means, if you sent the request via HTTPS, you will receive the answer via HTTPS. If you sent the request by email SMTP, the answer will return by email as well.

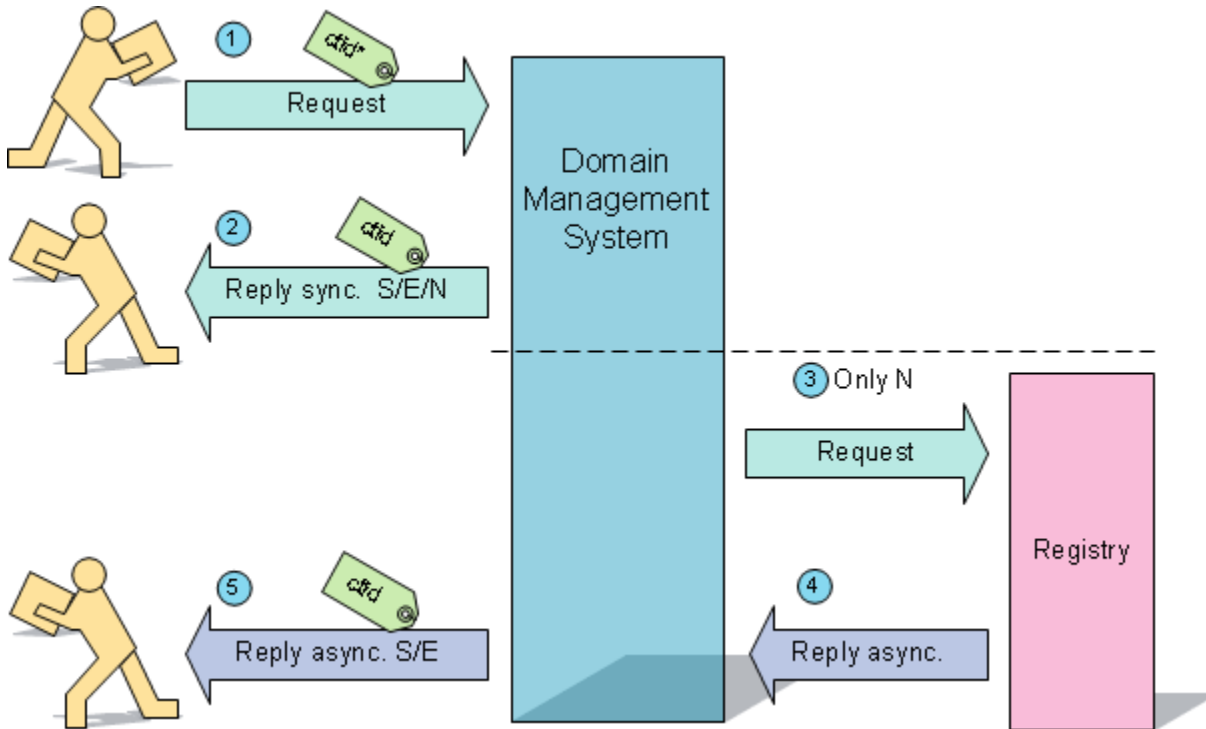
Asynchronous reply:

If you have received the reply N, the system will send you the reply solely by email (SMTP). At the end of the processing you will be informed about the status S or E.

The general rule

- all requests are confirmed by an asynchronous polling response (N)
- only for application of AuthInfo1 and AuthInfo2 and deletion of Authinfo1: synchronous confirmation (E or S)

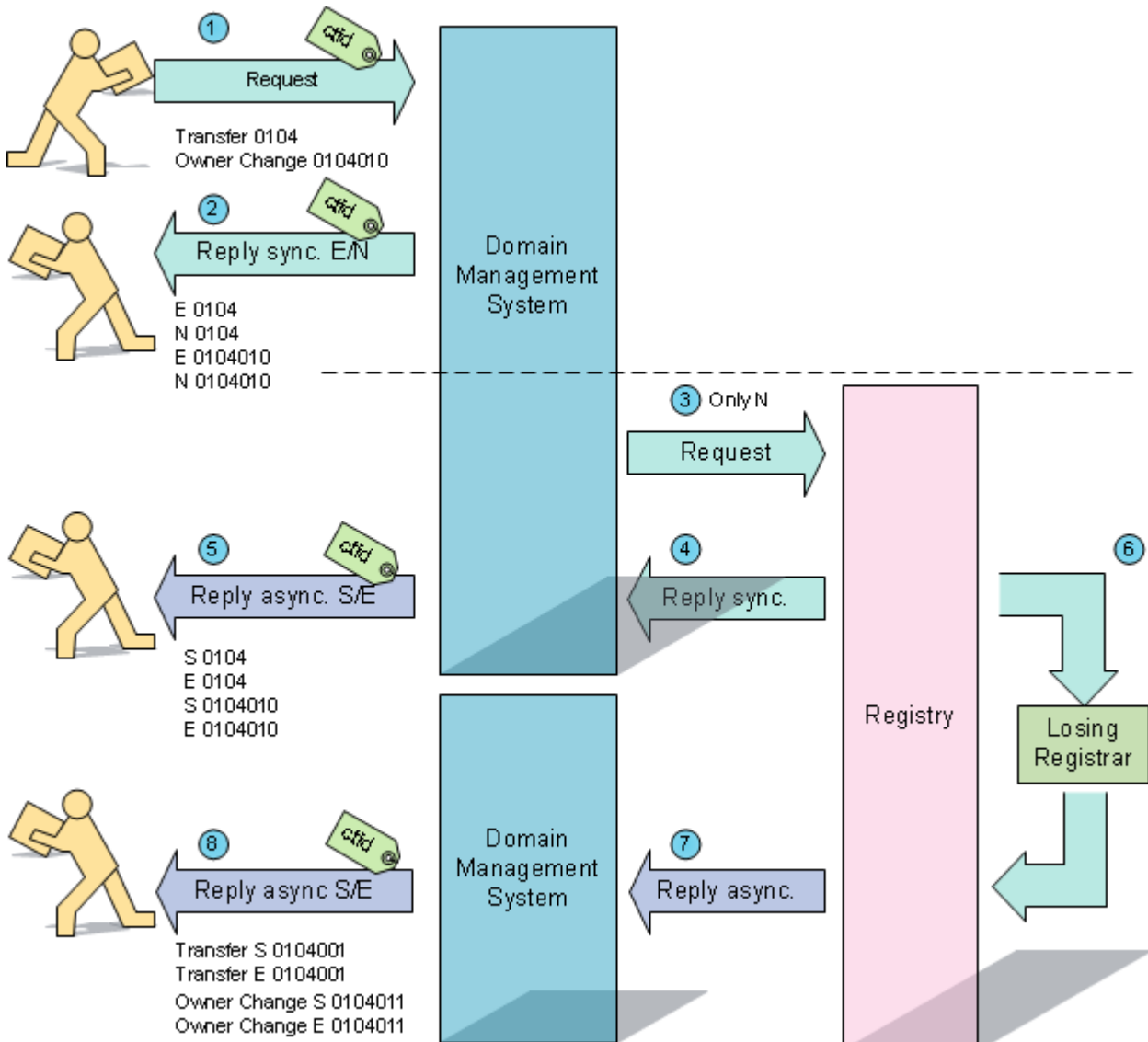
2.2 Domain and Domain Contact Transactions



**ctid= Client Transaktions ID

- ① You send your request to the domain management system.
- ② The domain management system immediately sends you a reply with the current status E if there is an error and S if it was successful. The status N is sent if the request is still being processed.
- ③ Requests that are answered with S or E are finished when the reply is sent. Requests that are answered with an N will be further processed by the system.
- ④ The domain management system sends the request to the registry and receives the answer.
- ⑤ The domain management system sends you a reply with the closing status code S or E.

2.3 Transfer-In Transactions and Owner Change



- 1 You send a transfer or owner change request to the domain management system.
- 2 The system immediately sends you a reply with the current status E if there is an error and S if it was successful. The status N is sent if the request is still being processed.
- 3 Requests that are answered with an N are sent to the registry for further processing.
- 4 The registry sends a reply to the system about the initialization of the transfer or owner change.
- 5 The domain management system sends you a reply with the current status S if it was successful or E if there was an error.

NOTE

If domain contact NIC-references are to be created, you will receive additional replies (S or E) of the type 0301 or 0302.

- ⑥ The registry sends the request to the losing registrar. The losing registrar has to answer the request.
- ⑦ The registry sends the answer to the domain management system.
- ⑧ The system sends you a reply with the closing status S or E for the transfer or owner change.

NOTE

For some TLDs the system additionally to the success message sends the following reply with the name of the new owner.

Example:

```
<system>
user: a3qs
task: 0104001
status: S0104001 [The incoming domain transfer (ACK) has been completed successfully.]
notice: MSG01050 [OwnerC has been imported.]
transaction: qs1258619858
last_transaction: qs1258619858
```

```
</system>
```

```
<reply>
tld: eu
sld: example
owner: a3qs
owner_context: 1
owner-c: 10431169
admin-c: 10431162
tech-c: 10431162
zone-c: 10431162
nserver: ns1.example.com
nserver: ns2.example.com
period: 1
status: success
authinfo:
```

```
</reply>
```

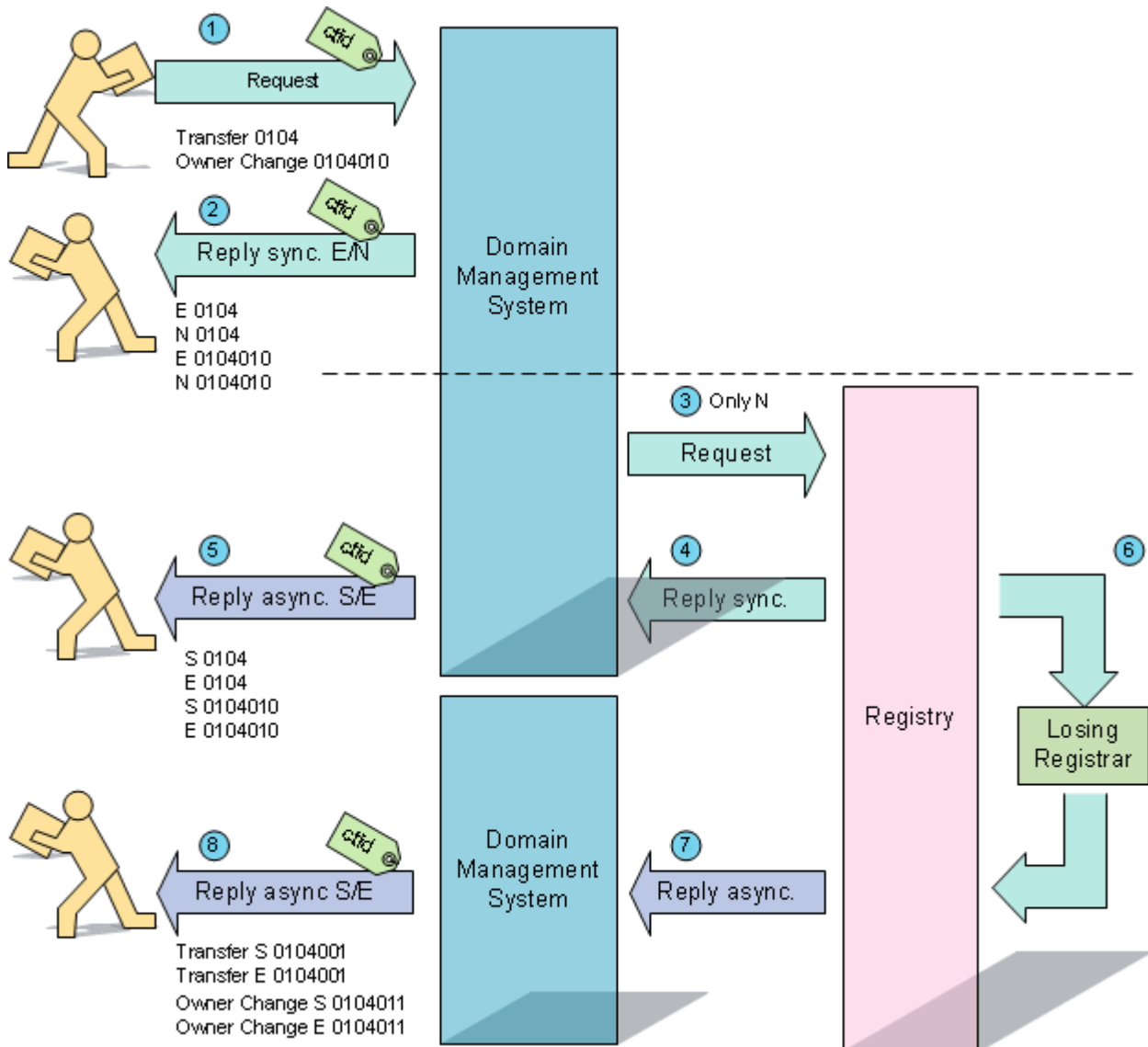
```
<notify>
...
</notify>
```

2.4 Transfer-Outs

a.) Transfer-Out with Registrar Involvement (gTLDs, .de)

NOTE

This is also the process for internal transfers.



- 1 The registry sends a notification to the domain management system, containing the information, that another provider requested for the transfer-out.
- 2 The system sends a reply, that the transfer was started (S1403).
- 3 You can send an acknowledgement for the transfer. (transfer_out_ack 0106). If you do not send it or if the transfer does not end with success, the automated process will continue.

④ If you do not send the transfer_out_ack (0106), you receive a reply (1403006 reminder) from the system, that you have to answer the request for transfer. If you don't answer again, the automated transfer process will continue.

NOTE

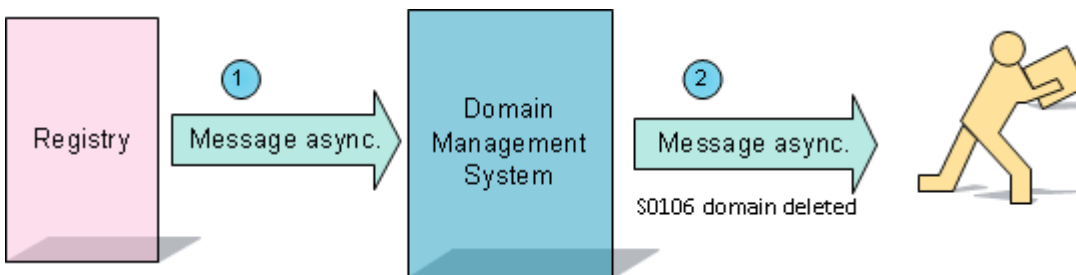
For gTLDs an AUTOAck is the standard practice.

⑤ After the transfer is processed, the system sends you a reply with the status of the transfer, e.g. S1403008, if the transfer was successful.

Examples

Reply Type	Code
1403 Transfer Start	<pre><system> user: sample_user task: 1403 status: S1403 [Outgoing transfer was started.] success_value: domain: example,com transaction: 1189217872 </system></pre>
	<p>NOTE The transfer data is in the reply block.</p>
0106 Transfer End	<pre><system> user: sample_user task: 0106 status: S0106 [Outgoing transfer was answered.] kk_answer: ACK </system></pre>
<p>1403008 Transfer-AutoACK</p> <p>You receive this reply when a transfer is not answered and the registry sends an Auto-ACK or when the transfer was not transferred out of our system. The domain will be removed from your inventory.</p>	<pre><system> user: sample_user task: 1403008 status: S1403008 [The domain was removed from your inventory.] success_value: domain: example.com transaction: 1189217877 </system></pre>

b.) Transfer-Out Replies without Registrar Involvement (.eu, .at)

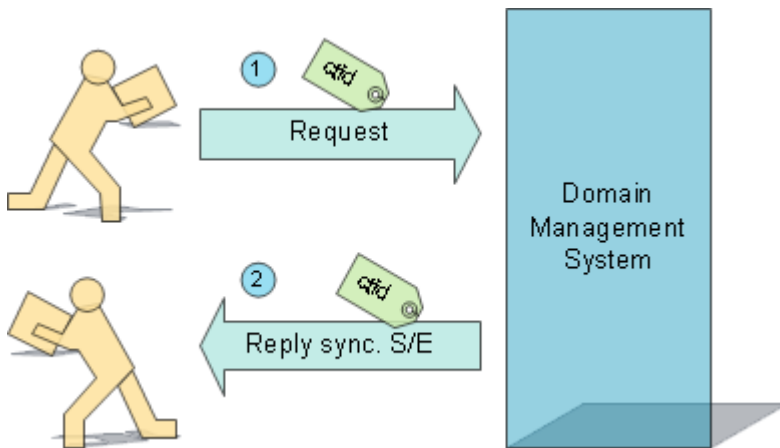


- ① The registry sends a reply to the system, containing the information, that an other provider requested for the transfer-out and that the domain owner agreed to the transfer.
- ② After the transfer is processed, the domain management system sends you a notification with the status of the transfer (e.g. S0106,.at, .eu), if the transfer was successful.

Examples

Reply Type	Code
0106 Transfer End	<pre><system> user: sample_user task: 0106 status: S0106 [Outgoing transfer was answered.] kk_answer: ACK </system></pre>

2.5 Zone, User and Inquire Transactions



- ① You send your request to the domain management system.
- ② The system sends you a notification with the closing status code S or E.

3 Polling

HINWEIS:

Die Funktion Polling muss von unserem Support freigeschaltet werden.

NOTE:

Polling has to be activated by our support.

3.1 General Information

Polling allows you to connect directly to our system and poll messages in XML format. If the messages are not polled within 24 hours, they are sent in XML format by email.

NOTE

Please keep in mind that we have to enable the polling function for you. Please contact our support.

Two types of messages are created by the system: polling job messages and polling notification messages.

Polling job messages are always sent to the user who has sent the request. Notifications, which are triggered by external events like the start of an outgoing transfer, are sent to the object owner.

3.1.1 The Receivers of Polling Job Messages and Polling Notifications

Polling job messages are always sent to the user who has sent the request. Notifications, which are triggered by external events like the start of an outgoing transfer, are sent to the object owner.

3.1.2 Retrieving the Messages in the Polling Queue

Two tasks in combination are used for retrieving messages in the polling queue: *PollInfo* and *PollACK*.

The first message is requested with the *PollInfo* task. This must then be confirmed with the *PollACK* task. The next message can then be requested, and also confirmed in turn. The XML task `<summary>` indicates the number of messages waiting in the polling queue. If a message is not polled within 24 hours, it is sent by email.

If the summary field returns the value "0", then this means that the data has not yet been processed by the system. Please execute the *PollInfo* task again in a few minutes.

NOTE

You cannot combine *PollInfo* and *PollAck* in a muti task request.

3.1.3 PollInfo- Task 0905

Mit dem Auftragstyp PollInfo rufen Sie die erste Benachrichtigung in der Warteschlange ab.

NOTE

It may take a few minutes for the *PollInfo* task to send a response.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>test</password>
    <context></context>
  </auth>
  <task>
    <code>0905</code>
  </task>
</request>
```

Example Response 1 with message block. Message was polled.

```
<response version="1">
  <result>
    <data>
      <summary>2</summary>
      <message>
        <id>1040</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <domain>
            <name>test-1292497929.eu</name>
            <ownerc>10458844</ownerc>
            <adminc>10458843</adminc>
            <techc>10458843</techc>
            <zonec>10458843</zonec>
            <nserver>
              <name>ns1.example.com </name>
            </nserver>
            <nserver>
              <name>ns2.example.com</name>
            </nserver>
            <period>1</period>
            <extension/>
            <owner>
              <user>customer</user>
              <context>104</context>
            </owner>
          </domain>
          <created>2010-12-16 12:59:51</created>
          <nic_response></nic_response>
          <status>
            <text>Domain has been registered successfully.</text>
            <code>S0101</code>
            <type>success</type>
          </status>
          <ctid>379</ctid>
        </job>
      </message>
    </data>
```

```

    <!-- Status des Poll Request -->
    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20101217-TEST-41</stid>
</response>

```

Example Response 2: No polling information is available at the moment

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>0</summary>
    </data>
    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20101217-TEST-39674</stid>
</response>

```

3.1.4 PollAck-Task 0906

To acknowledge a polling message, you have to send the task 0906 including the respective message ID to the system.

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0906</code>
    <message>
      <id>1</id>
    </message>
  </task>
</request>

```

Example: Response 1 with messages in queue

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>2</summary>
      <message>
        <id>1152</id>
      </message>
    </data>
    <status>
      <code>S0906</code>
      <text>System message was confirmed successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

```

    </status>
  </result>
  <stid>20101221-TEST-58150</stid>
</response>

```

Example: Response 2 without messages in queue

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>0</summary>
    </data>
    <status>
      <code>S0906</code>
      <text>System message was confirmed successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20101221-TEST-58150</stid>
</response>

```

3.2 Messages

3.2.1 Polling Job Messages

Polling job messages are the answer of the system concerning a request. All polling job messages include the status and the message ID of the request. So it is possible to inquire the status of the request.

There are following status:

- success: the request was carried out successfully.
- error: the request has not been carried out successfully
- notify : the request is still in process (preliminary report e.g for transfers)

3.2.2 Polling Notification Messages

The following notification messages are triggered by external events like an outgoing domain transfer.

Transfer-Out

- *transfer_start*: start of an outgoing domain transfer
- *transfer_autoack*: the outgoing domain transfer was acknowledged automatically. (AutoAck). Only for gTLDs.
- *transfer_automack*: the outgoing domain transfer was refused automatically (AutoNack).
- *transfer_reminder*: reminder for an ongoing transfer
- *domain_away*: the domain was deleted from your inventory (e.g. for transfers without registrar involvement)

Domain Auto Update

- *autoupdate_dns_error*: the automatic update was successfully carried out (for .de domains). An automatic update is only then carried out when the connectivity check that is performed during the registration procedure fails for any reason.
- *autoupdate_dns_success*:
Check for connectivity was successful.

OCVAL (OwnerC validation)

- *ocval_reminder*: *Reminder, that the domain will be disconnected in 5 days.*
- *ocval_activation*: Information that the nameservers have been disconnected and the domain is not available.
- *ocval_deactivation*: information that the domain has been reconnected. The domain is reachable again.

Structure of the Polling Notification Messages

3.3 Domain Jobs

3.3.1 Domain Create. Examples for Polling Notifications

Response S0101: job success

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Antwort version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21593</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <domain>
            <name>example.com</name>
            <ownerc>10465800</ownerc>
            <adminc>10465800</adminc>
            <techc>10465800</techc>
            <zonec>10465800</zonec>
            <nserver>
              <name>ns1.example.com</name>
            </nserver>
            <nserver>
              <name>ns2.example.com</name>
            </nserver>
            <period>1</period>
            <authinfo>819fd1d6b1</authinfo>
            <owner>
              <user>customer</user>
              <context></context>
            </owner>
          </domain>
          <status>
```

```

        <text>Domain has been registered successfully.</text>
        <code>S0101</code>
        <type>success</type>
        <object>
            <type>domain</type>
            <value>example.com</value>
        </object>
    </status>
    <ctid>616</ctid>
</job>
<created>2011-05-27 12:38:05</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110527-app1-test-43393</stid>
</response>

```

Response E0101: Job failed

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Antwort version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21592</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <domain>
            <name>example.com</name>
          </domain>
          <nic_Antwort></nic_Antwort>
          <status>
            <text>Domain could not be registered.</text>
            <code>E0101</code>
            <type>error</type>
            <object>
              <type>domain</type>
              <value>example.com</value>
            </object>
          </status>
          <msg>
            <text>The domain is already registered.</text>
            <code>EF01021</code>
            <type>error</type>
          </msg>
          <ctid>613</ctid>
        </job>
        <created>2011-05-27 12:02:55</created>
      </message>
    </data>
    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20110527-app1-43397</stid>
</response>

```

3.3.2 Domain Update. Examples for Polling Notifications

Response 1 S0102: Job success (asychronal polling answer)

```
<?xml version="1.0"encoding="UTF-8"standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21600</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <domain>
            <name>example.com</name>
            <ownerc>10465806</ownerc>
            <adminc>10465806</adminc>
            <techc>10465806</techc>
            <zonec>10465806</zonec>
            <nserver>
              <name>ns1.example.com</name>
            </nserver>
            <nserver>
              <name>ns2.example.com</name>
            </nserver>
            <period>1</period>
            <authinfo>6d5a0d44b6</authinfo>
            <owner>
              <user>customer</user>
              <context></context>
            </owner>
          </domain>
          <status>
            <text>The domain has been updated successfully.</text>
            <code>S0102</code>
            <type>success</type>
            <object>
              <type>domain</type>
              <value>example.com</value>
            </object>
          </status>
          <ctid>625</ctid>
        </job>
        <created>2011-05-27 13:49:58</created>
      </message>
    </data>
    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20110527-appl-test-43762</stid>
</response>
```

Response 2 S0102: Job failed (asychronal polling answer)

```
<?xml version="1.0"encoding="UTF-8"standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
```

```

<id>21598</id>
<owner>
  <user>customer</user>
  <context></context>
</owner>
<job>
  <domain>
    <name>example.com</name>
    <ownerc>10465805</ownerc>
    <adminc>10465805</adminc>
    <techc>10465805</techc>
    <zonec>10465805</zonec>
    <nserver>
      <name>ns1.example.com</name>
    </nserver>
    <nserver>
      <name>ns2.example.com</name>
    </nserver>
    <period>1</period>
    <authinfo>ccde023c9c</authinfo>
    <owner>
      <user>customer</user>
      <context>4</context>
    </owner>
  </domain>
  <nic_response></nic_response>
  <status>
    <text>Domain could not be updated./</text>
    <code>E0102</code>
    <type>error</type>
    <object>
      <type>domain</type>
      <value>example.com</value>
    </object>
  </status>
  <msg>
    <text>The object cannot be updated due to the domain
status.</text>
    <code>EF00714</code>
    <type>error</type>
  </msg>
  <ctid>623</ctid>
</job>
<created>2011-05-27 13:46:04</created>
</message>
</data>
<status>
  <code>S0905</code>
  <text>System message was polled successfully.</text>
  <type>success</type>
</status>
</result>
<stid>20110527-app1-test-43666</stid>
</response>

```

3.3.3 Domain Ownerchange. Examples for Polling Notifications

Response 1 E0104010: Job failed - *transfer status failed*

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21656</id>

```

```

    <owner>
      <user>customer</user>
      <context></context>
    </owner>
    <job>
      <domain>
        <name>example.com</name>
      </domain>
      <nic_response></nic_response>
      <status>
        <text>Domain owner change could not be completed.</text>
        <code>E0104010</code>
        <type>error</type>
        <object>
          <type>domain</type>
          <value>example.com</value>
        </object>
      </status>
      <msg>
        <text>Ungueltige AuthInfo</text>
        <code>EF00804</code>
        <type>error</type>
      </msg>
      <ctid>659</ctid>
    </job>
    <created>2011-05-27 17:12:16</created>
  </message>
</data>
<status>
  <code>S0905</code>
  <text>System message was polled successfully.</text>
  <type>success</type>
</status>
</result>
<stid>20110527-appl-test-47367</stid>
</response>

```

Response 2 S0104010: Job pending - transfer status start

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21657</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <transfer>
            <status>start</status>
          </transfer>
          <domain>
            <name>example.com</name>
          </domain>
          <status>
            <text>Owner change has been started successfully.</text>
            <code>S0104010</code>
            <type>notify</type>
            <object>
              <type>domain</type>
              <value>example.com</value>
            </object>
          </status>
          <ctid>660</ctid>
        </job>
      </message>
    </data>
  </result>
</response>

```



```

        </job>
        <created>2011-05-27 17:14:45</created>
    </message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110527-appl-test-47426</stid>
</response>

```

Response 3 E0104011: Job failed- transfer status *nack*

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<response version="1">
    <result>
        <data>
            <summary>5</summary>
            <message>
                <id>118887</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <job>
                    <transfer>NACK</transfer>
                    <domain>
                        <name>example.com</name>
                    </domain>
                    <nic_response></nic_response>
                    <status>
                        <text>Incoming owner change could not be completed successfully
                        (NACK).</text>
                        <code>E0104011</code>
                        <type>error</type>
                        <object>
                            <type>domain</type>
                            <value>test-31102011.com</value>
                        </object>
                    </status>
                    <ctid>664</ctid>
                </job>
                <created>2011-10-31 16:00:09</created>
            </message>
        </data>
    </result>
    <status>
        <code>S0905</code>
        <text>System message was polled successfully.</text>
        <type>success</type>
    </status>
</result>
</response>

```

Response 4: job pending - transfer status *ack*

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<response>
    <result>
        <data>
            <summary>1</summary>
            <message>
                <id>21664</id>
                <owner>
                    <user>customer</user>
                    <context></context>

```

```

    </owner>
    <job>
      <transfer>
        <status>ack</status>
      </transfer>
      <domain>
        <name>example.com</name>
        <ownerc>10465839</ownerc>
        <adminc>10465838</adminc>
        <techc>10465838</techc>
        <zonec>10465838</zonec>
        <registry_status>ACTIVE</registry_status>
        <nserver>
          <name>ns1.example.com</name>
        </nserver>
        <nserver>
          <name>ns2.example.com</name>
        </nserver>
        <period>10</period>
        <authinfo>30a480fdc0</authinfo>
        <owner>
          <user>customer</user>
          <context>1</context>
        </owner>
      </domain>
      <status>
        <text>Incoming owner change (ACK) has been completed
        successfully.</text>
        <code>S0104011</code>
        <type>notify</type>
        <object>
          <type>domain</type>
          <value>example.com</value>
        </object>
      </status>
      <ctid>662</ctid>
    </job>
    <created>2011-05-27 17:28:20</created>
  </message>
</data>
<status>
  <code>S0905</code>
  <text>System message was polled successfully.</text>
  <type>success</type>
</status>
</result>
<stid>20110527-app1-test-47588</stid>
</response>

```

Response 5 S0102: Job success - transfer status *autoupdate_success*

```

<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21665</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
      <job>
        <transfer>
          <status>autoupdate_success</status>
        </transfer>
      <domain>

```

```

<name>example.com</name>
<ownerc>10465839</ownerc>
<adminc>10465838</adminc>
<techc>10465838</techc>
<zonec>10465838</zonec>
<registry_status>ACTIVE</registry_status>
<nserver>
  <name>ns1.example.com</name>
</nserver>
<nserver>
  <name>ns2.example.com</name>
</nserver>
<period>10</period>
<authinfo>b24f92f397</authinfo>
<owner>
  <user>customer</user>
  <context>
    </context>
  </owner>
</domain>
<status>
  <text>The domain has been updated successfully.</text>
  <code>S0102</code>
  <type>success</type>
  <object>
    <type>domain</type>
    <value>example.com</value></object>
  </status>
  <ctid>662</ctid>
</job>
<created>2011-05-27 17:28:32</created>
</message>
</data>
<status>
  <code>S0905</code>
  <text>System message was polled successfully.</text>
  <type>success</type>
</status>
</result>
<stid>20110527-appl-test-47599</stid>
</response>

```

Response 6 E0102: Job success - Transfer status *transfer autoupdate_failed*

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21668</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <transfer>
            <status>autoupdate_failed</status>
          </transfer>
          <domain>
            <name>example.com</name>
            <ownerc>10465841</ownerc>
            <adminc>10465840</adminc>
            <techc>10465840</techc>
            <zonec>10465840</zonec>
            <registry_status>ACTIVE</registry_status>
            <nserver>

```

```

        <name>ns1.example.com</name>
    </nserver>
    <nserver>
        <name>ns2.example.com</name>
    </nserver>
    <period>10</period>
    <authinfo>a7a15b8d76</authinfo>
    <owner>
        <user>customer</user>
        <context>1</context>
    </owner>
</domain>
<nic_response></nic_response>
<status>
    <text>The domain could not be updated successfully.</text>
    <code>E0102</code>
    <type>success</type>
    <object>
        <type>domain</type>
        <value>example.com</value>
    </object>
</status>
<msg>
    <text>The object cannot be updated due to the domain
    status.</text>
    <code>EF00714</code>
    <type>error</type>
</msg>
    <ctid>663</ctid>
</job>
    <created>2011-05-27 17:31:02</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
    <stid>20110527-app1-test-47664</stid>
</response>

```

3.4 Domain Transfer Jobs

3.4.1 Domain TransferIn: Examples for Polling Notifications

Response 1: E0104: Job failed - transfer status *failed*

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>1</summary>
            <message>
                <id>21627</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <job>
                    <transfer>
                        <status>failed</status>
                    </transfer>
                    <domain>

```

```

        <name>example.com</name>
    </domain>
    <nic_response></nic_response>
    <status>
        <text>Domain transfer could not be carried out
        successfully.</text>
        <code>E0104</code>
        <type>error</type>
        <object>
            <type>domain</type>
            <value>example.com</value>
        </object>
    </status>
    <msg>
        <text>Invalid AuthInfo</text>
        <code>EF00804</code>
        <type>error</type>
    </msg>
    <ctid>644</ctid>
</job>
<created>2011-05-27 15:05:20</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid></stid>
    20110527-app1-test-44671
</response>

```

Response 2: S0104: Job pending - transfer status start

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>3</summary>
            <message>
                <id>21628</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <job>
                    <transfer>
                        <status>start</status>
                    </transfer>
                    <domain>
                        <name>example.com</name>
                    </domain>
                    <status>
                        <text>Domain transfer has been started successfully.</text>
                        <code>S0104</code>
                        <type>notify</type>
                        <object>
                            <type>domain</type>
                            <value>example.com</value>
                        </object>
                    </status>
                    <ctid>645</ctid>
                </job>
                <created>2011-05-27 15:08:29</created>
            </message>
        </data>
    </result>
</response>

```

```

    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20110527-app1-test-44754</stid>
</response>

```

Response 3: E0104000:1 Job failed- transfer status NACK

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21635</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <transfer>
            <status>nack</status>
          </transfer>
          <domain>
            <name>example.com</name>
          </domain>
          <status>
            <text>Incoming domain transfer (NACK) could not be completed
            successfully.</text>
            <code>E0104001</code>
            <type>error</type>
            <object>
              <type>domain</type>
              <value>example.com</value>
            </object>
          </status>
          <ctid>648</ctid>
        </job>
        <created>2011-05-27 15:17:42</created>
      </message>
    </data>
    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20110527-app1-test-44977</stid>
</response>

```

Response 4: S0104001: Job pending - transfer status ACK

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>2</summary>
      <message>
        <id>21629</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <transfer>

```

```

        <status>ack</status>
    </transfer>
    <domain>
        <name>example.com</name>
        <ownerc>10465818</ownerc>
        <adminc>10465818</adminc>
        <techc>10465818</techc>
        <zonec>10465818</zonec>
        <registry_status>ACTIVE</registry_status>
        <nserver>
            <name>ns1.example.com</name>
        </nserver>
        <nserver>
            <name>ns2.example.com</name>
        </nserver>
        <period>10</period>
        <authinfo>4e0f4c7b33</authinfo>
        <owner>
            <user>customer</user>
            <context></context>
        </owner>
    </domain>
    <status>
        <text>The incoming domain transfer (ACK) has been completed
        successfully.</text>

        <code>S0104001</code>
        <type>notify</type>
        <object>
            <type>domain</type>
            <value>example.com</value>
        </object>
    </status>
    <ctid>645</ctid>
</job>
<created>2011-05-27 15:08:41</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110527-appl-test-44757</stid>
</response>

```

Response 5: S0102: Job success - transfer status *autoupdate_success* or transfer *autoupdate_failed*

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>1</summary>
            <message>
                <id>21630</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <job>
                    <transfer>
                        <status>autoupdate_success</status>
                    </transfer>
                    <domain>
                        <name>example.com</name>

```

```

        <ownerc>10465818</ownerc>
        <adminc>10465818</adminc>
        <techc>10465818</techc>
        <zonec>10465818</zonec>
        <registry_status>ACTIVE</registry_status>
        <nserver>
            <name>ns1.example.com</name>
        </nserver>
        <nserver>
            <name>ns2.example.com</name>
        </nserver>
        <period>10</period>
        <authinfo>e12023cafb</authinfo>
        <owner>
            <user>customer</user>
            <context>4</context>
        </owner>
    </domain>
    <status>
        <text>The domain has been updated successfully.</text>
        <code>S0102</code>
        <type>success</type>
        <object>
            <type>domain</type>
            <value>example.com</value>
        </object>
    </status>
    <ctid>645</ctid>
</job>
<created>2011-05-27 15:08:50</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110527-appl-test-44760</stid>
</response>

```

Response 6: E0102: Job success - Transfer status *transfer autoupdate_failed*

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>1</summary>
            <message>
                <id>21668</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <job>
                    <transfer>
                        <status>autoupdate_failed</status>
                    </transfer>
                    <domain>
                        <name>example.com</name>
                        <ownerc>10465841</ownerc>
                        <adminc>10465840</adminc>
                        <techc>10465840</techc>
                        <zonec>10465840</zonec>
                        <registry_status>ACTIVE</registry_status>
                        <nserver>
                            <name>ns1.example.com</name>

```



```

        </nserver>
        <nserver>
            <name>ns2.example.com</name>
        </nserver>
        <period>10</period>
        <authinfo>a7a15b8d76</authinfo>
        <owner>
            <user>customer</user>
            <context>4</context>
        </owner>
    </domain>
    <nic_response></nic_response>
    <status>
        <text>Domain could not be updated.</text>
        <text>/text>
        <code>E0102</code>
        <type>success</type>
        <object>
            <type>domain</type>
            <value>example.com</value>
        </object>
    </status>
    <msg>
        <text>The object cannot be updated due to the domain
        status.</text>
        <code>EF00714</code>
        <type>error</type>
    </msg>
    <ctid>663</ctid>
</job>
<created>2011-05-27 17:31:02</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110527-app1-test-47664</stid>
</response>

```

3.4.2 Domain Transfer Out. Examples for Polling Notifications

Response 1: S1403 Transfer out start

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>2</summary>
            <message>
                <id>21672</id>
                <owner>
                    <user>
                        customer/user>
                    <context>4</context>
                </owner>
                <notify>
                    <type>transfer_start</type>
                    <domain>
                        <name>example.com</name>
                    </domain>
                    <created>2011-05-30 10:44:09</created>
                    <transfer>
                        <gaining_registrar>OTHER-REG</gaining_registrar>

```

```

        <losing_registrar>EXAMPLE-REG</losing_registrar>
        <start>2011-05-30 10:44:07</start>
        <reminder>2011-05-30 10:54:07</reminder>
        <autonack>2011-05-30 11:04:07</autonack>
        <end>2011-05-30 11:14:07</end>
    </transfer>
    <status>
        <text>Outgoing domain transfer has been started
        successfully.</text>
        <code>S1403</code>
        <type>success</type>
    </status>
</notify>
<created>2011-05-30 10:44:09</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110530-appl-test-173</stid>
</response>

```

Answer 1 Polling: TransferOut Auto ACK S1403006

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>6</summary>
            <message>
                <id>978</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <notify>
                    <type>transfer-autoack</type>
                    <domain>
                        <name>adns1291883804.info</name>
                    </domain>
                    <created>2010-12-13 12:37:31</created>
                    <transfer>
                        <gaining_registrar>AFILIAS-TEST</gaining_registrar>
                        <start>2010-12-09 08:37:30</start>
                        <reminder>2010-12-12 08:37:30</reminder>
                        <autonack>2010-12-13 12:37:30</autonack>
                        <end>2010-12-14 08:37:30</end>
                    </transfer>
                    <status>
                        <text>.Reminder! Outgoing transfer was started.</text>
                        <code>S1403006</code>
                        <type>success</type>
                    </status>
                </notify>
            </message>
        </data>
        <status>
            <code>S0905</code>
            <text>System message was polled successfully.</text>
            <type>success</type>
        </status>
    </result>
    <stid>20101213-TEST-45653</stid>
</response>

```

Response 4: S1403005 Transfer Out AutoNACK

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1"
  <result>
    <data>
      <summary>2</summary>
      <message>
        <id>21762</id>
        <owner>
          <user>customer</user>
          <context>4</context>
        </owner>
        <notify>
          <type>transfer_autonack</type>
          <domain>
            <name>example.com</name>
          </domain>
          <created>2011-06-01 10:52:06</created>
          <transfer>
            <gaining_registrar>OTHER-REG</gaining_registrar>
            <start>2011-06-01 10:49:45</start>
            <reminder>2011-06-01 10:50:45</reminder>
            <autonack>2011-06-01 10:51:45</autonack>
            <end>2011-06-01 10:52:45</end>
          </transfer>
          <status>
            <text>The request was not answered in time AUTO-NACK was
              processed.</text>
            <code>S1403005</code>
            <type>success</type>
          </status>
        </notify>
        <created>2011-06-01 10:52:06</created>
      </message>
    </data>
    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20110601-appl-test-32548</stid>
</response>

```

Response 2: S1403006 Transfer out reminder for running transfers

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>21761</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <notify>
          <type>transfer_reminder</type>
          <domain>
            <name>example.com</name>
          </domain>
          <created>2011-06-01 10:51:03</created>
          <transfer>
            <gaining_registrar>OTHER-REG</gaining_registrar>
            <start>2011-06-01 10:49:45</start>

```

```

        <reminder>2011-06-01 10:50:45</reminder>
        <autonack>2011-06-01 10:51:45</autonack>
        <end>2011-06-01 10:52:45</end>
    </transfer>
    <status>
        <text>Reminder! Outgoing transfer was started.</text>
        <code>S1403006</code>
        <type>success</type>
    </status>
</notify>
<created>2011-06-01 10:51:03</created>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110601-app1-test-32511</stid>
</response>

```

Answer 2 Polling: Domain Away S0106

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>3</summary>
      <message>
        <id>486721</id>
        <owner>
          <user>e2167f5b-2973-464d-9fbc-3c495dalc428</user>
          <context>4</context>
        </owner>
        <job>
          <domain>
            <name>xxxxx-transfer-polling.com</name>
          </domain>
          <answer>ACK</answer>
          <status>
            <text>Outgoing domain transfer has been answered
            successfully.</text>
            <code>S0106</code>
            <type>success</type>
            <object>
              <type>domain</type>
              <value>xxxxx-transfer-polling.com</value>
            </object>
          </status>
        </job>
        <created>2016-05-30 10:26:45</created>
      </message>
    </data>
    <status>
      <code>S0905</code>
      <text>The notification was polled successfully.</text>
      <type>success</type>
      <object>
        <type>message</type>
        <value>486721</value>
      </object>
    </status>
  </result>
  <stid>20160530-app1-dev-1274</stid>
</response>

```

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>1047</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <notify>
          <type>domain-away</type>
          <domain>
            <name>example.com</name>
          </domain>
          <created>2010-12-17 19:02:06</created>
          <transfer/>
          <status>
            <text>Outgoing domain transfer has been answered
            successfully.</text>
            <code>S0106</code>
            <type>success</type>
          </status>
          <msg>
            <code>MSG010409</code>
            <type>notice</type>
            <text>The registry removed the domain from your inventory.
            You are no longer the OwnerC of the domain.</text>
          </msg>
        </notify>
      </message>
    </data>
    <status>
      <code>S0905</code>
      <text>System message was polled successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20101217-TEST-65632</stid>
</response>

```

3.5 Contact Jobs

3.5.1 Contact Create. Examples for Polling Notifications

Response 1: E0301:job failed

```

xml version="1.0" encoding="utf-8" ?>
<response version="1">
  <result>
    <data>
      <summary>1</summary>
      <message>
        <id>2166684</id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <job>
          <handle>
            <type>PERSON</type>
            <fname>Peter</fname>
            <lname>Doe</lname>
          </handle>
        </job>
      </message>
    </data>
  </result>
</response>

```

```

        <title></title>
        <organization>PDA</organization>
        <address>Example Street 3</address>
        <pcode>12345</pcode>
        <city>Example City</city>
        <state></state>
        <country>USA</country>
        <phone>+49-12345-12345</phone>
        <fax>+49-12345-12345</fax>
        <email>Doe@example.com</email>
        <sip></sip>
        <protection>B</protection>
        <nic_ref>de</nic_ref>
        <nic_ref>com</nic_ref>
        <nic_ref>net</nic_ref>
        <remarks></remarks>
    </handle>
    <nic_response></nic_response>
    <status>
        <text>Domain contact could not be created.</text>
        <code>E0301</code>
        <type>error</type>
        <object>
            <type>handle</type>
            <value>10544500</value>
        </object>
    </status>
    <ctid>qa-96123-2360</ctid>
</job>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20111102-appl-test-52285</stid>
</response>

```

Response 2: S0301 Job ok

```

<xml version="1.0" encoding="utf-8" ?>
  <response version="1">
    <result>
      <data>
        <summary>2</summary>
        <message>
          <id>21684</id>
          <owner>
            <user>customer</user>
            <context></context>
          </owner>
          <job>
            <handle>
              <type>PERSON</type>
              <fname>Peter</fname>
              <lname>Doe</lname>
              <title></title>
              <organization>PDA</organization>
              <address>Example Street 3</address>
              <pcode>12345</pcode>
              <city>Example City</city>
              <state></state>
              <country>USA</country>
              <phone>+49-12345-12345</phone>
              <fax>+49-12345-12345</fax>
            </handle>
          </job>
        </message>
      </data>
    </result>
  </response>

```

```

        <email>Doe@example.com</email>
        <protection>B</protection>
        <remarks></remarks>
        <extension/>
        <owner>
            <user>customer</user>
            <context></context>
        </owner>
        <id>10465849</id>
        <created>2011-05-30 14:40:52</created>
    </handle>
    <status>
        <text>Domain contact has been successfully
        created.</text>
        <code>S0301</code>
        <type>success</type>
        <object>
            <type>handle</type>
            <value>10465849</value>
        </object>
    </status>
    <ctid>448</ctid>
</job>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110530-app1-test-1283</stid>
</response>

```

3.5.2 Contact Update. Examples for Polling Notifications

Response 1: E0302 Job failed

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>1</summary>
            <message>
                <id>21802</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <job>
                    <handle>
                        <type>PERSON</type>
                        <fname>Peter</fname>
                        <lname>Doe</lname>
                        <title></title>
                        <organization>PDA</organization>
                        <address>Example Street 3</address>
                        <pcode>12345</pcode>
                        <city>Example City</city>
                        <state></state>
                        <country>USA</country>
                        <phone>+49-12345-12345</phone>
                        <fax>+49-12345-12345</fax>
                        <email>Doe@example.com</email>
                        <sip></sip>
                        <email>customer@example.com</email>
                    </handle>
                </job>
            </message>
        </data>
    </result>
</response>

```

```

        <protection>B</protection>
        <remarks></remarks>
        <nic_ref>
            <type>ALL</type>
            <role>ALL</role>
            <nic>de</nic>
            <status>failed</status>
            <name></name>
        </nic_ref>
        <extension/>
        <owner>
            <user>customer</user>
            <context>l</context>
        </owner>
        <id>10465899</id>
        <created>2011-06-01 14:29:01</created>
    </handle>
    <nic_response></nic_response>
    <status>
        <text>Domain contact could not be updated.</text>
        <code>E0302</code>
        <type>error</type>
        <object>
            <type>handle</type>
            <value>10465899</value>
        </object>
    </status>
    <ctid>2223</ctid>
</job>
</message>
</data>
<status>
    <code>S0905</code>
    <text>System message was polled successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20110601-appl-test-44949</stid>
</response>

```

Response 2: S0302 Job ok

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
    <result>
        <data>
            <summary>l</summary>
            <message>
                <id>21797</id>
                <owner>
                    <user>customer</user>
                    <context></context>
                </owner>
                <job>
                    <handle>
                        <type>PERSON</type>
                        <fname>Peter</fname>
                        <lname>Doe</lname>
                        <title></title>
                        <organization>PDA</organization>
                        <address>Example Street 3</address>
                        <pcode>12345</pcode>
                        <city>Example City</city>
                        <state></state>
                        <country>USA</country>
                        <phone>+49-12345-12345</phone>
                        <fax>+49-12345-12345</fax>
                    </handle>
                </job>
            </message>
        </data>
    </result>
</response>

```



```
<email>Doe@example.com</email>
<protection>B</protection>
<remarks></remarks>
<nic_ref>
  <type>ALL</type>
  <role>ALL</role>
  <nic>...</nic>
  <status>success</status>
  <name></name>
</nic_ref>
<extension/>
<owner>
  <user>customer</user>
  <context>1</context>
</owner>
<id>10465897</id>
<created>2011-06-01 12:41:53</created>
</handle>
  <status>
    <text>Domain contact has been updated
    successfully.</text>
    <code>S0302</code>
    <type>success</type>
    <object>
      <type>handle</type>
      <value>10465897</value>
    </object>
  </status>
  <ctid>2223</ctid>
</job>
</message>
</data>
<status>
  <code>S0905</code>
  <text>System message was polled successfully.</text>
  <type>success</type>
</status>
</result>
<stid>20110601-appl-test-44868</stid>
</response>
```

4 Domain

4.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	Domain Create 0101 Domain CreatePlus 0101 Domain Update 0102 Domain UpdatePlus 0102 Domain Renew 0101003 Domain Ownerchange 0104010 Domain Delete 0103 Domain List 0105 Domain Status 0102002 Domain Premium Info 0164 Domain Premium List 0165
domain	1 - n		
domain / name	1	string	Domain name
domain / ignore_whois	0 - 1		Value 1 = The availability check is being bypassed.
domain / owncrc	1	contact	Owner contact ID or the complete contact block
domain / adminc	0 - 1*	contact	Administrative contact ID or the complete contact block
domain / techc	0 - 1*	contact	Technical contact ID or the complete contact block
domain / zonec	0 - 1*	contact	Zone contact ID or the complete contact block
domain / period	0 - 1*	int	Registration period. Default is the minimum registration period; the registration will be renewed automatically.
domain / nserver	0 - 8*		
domain / nserver / name	1	domain	Host name of the nameserver
domain / nserver / ip	0 - 1*	ip	Nameserver IP, only for GLUE records; in this case it is mandatory. NOTE GLUE records end with the domain name. In case of using GLUE records you have to send the <ip> element containing the IPv4-address of the nameserver additionally. Please note that for some registries the GLUE records will be removed from the request, because the registries assume registered domains. In this cases you have to send a <i>DomainUpdate</i> request to add nameservers.
domain / nserver / ip6	0 - 1*	ip6	IPv6 of the nameserver, only for GLUE records; optional. NOTE: The nameserver IPv6 can only be entered, if an IPv4 value is entered too.

Key	Occurs	Format	Description
domain / nsenry	0 - 5*	nsenry	NSentry is only be provided for .DE. If NSentry is used, nserver is not allowed. In a bulk request nsenry is only allowed in the "default"-block. The domain has to be substituted with the placeholder {domain}.
domain / nscheck	0 - 1	enum	An additional nameserver check will be carried out. <ul style="list-style-type: none"> • soa = only soa values will be checked • ns = only nameservers will be checked • all = soa and ns will be checked
domain / zone / ip	0 - 1		IP address of the zone (A Record)
domain / zone / mx	0 - n		MX record (mailserver) . Enter the complete domain host name of the mailserver.
domain / zone / ns_action	0 - 1	string	<p><u>Nameserver-Modus</u></p> <ul style="list-style-type: none"> • complete: the zone will be generated on all nameservers • primary: the zone will only be generated on the primary nameserver • secondary: the zone will only be generated on the secondary nameserver. Zone data will be transferred to the primary nameserver via AXFR • hidden primary: the zone will be generated on the primary nameserver. The zone data will be transferred to the secondary nameserver via AXFR. The primary nameserver is not displayed as a nameserver for the zone.
domain / zone / www_include	0 - 1	enum	Automatic creation of the "www" subdomain. <ul style="list-style-type: none"> ▪ 1 = the system automatically creates the "www" subdomain and assigns it to the main IP address of the domain. ▪ 0 = no www-subdomain will be created.
domain / registry_status	0 - 1	enum	Sets the domain status after the "Create". <p><u>The statuses</u></p> <ul style="list-style-type: none"> • LOCK = the domain is protected against an outgoing transfer. • HOLD = the domain is registered but not connected. (Example: resellers can set this status e.g. if a client did not pay.) • HOLD-LOCK= the domain is registered but not connected. It is also protected against an outgoing transfer. (Example: resellers can set this status e.g. if a client did not pay.) • ACTIVE = domain will be unlocked. <p>Default: LOCK</p>
domain / use_trustee	0 - 1	enum	Trustee service 1= true You can also enter the word "true" as a valid value.

Key	Occurs	Format	Description
domain / use_privacy	0 - 1	enum	Usage of the Whois Privacy Service for the domain. Using Privacy Service your contact data will not appear in the public Whois. 1= true You can also enter the word "true" as a valid value. The key is optional and only available for some domains. You can use the task <i>user inquire</i> to inquire the respective tasks.
domain / confirm_order	0 - 1	int	Some TLDs require a confirmation of the "Terms and Conditions of Registration". 1 = confirmation For other TLDs the key will be ignored.
domain / execdate	0 - 1	date time	execution date. Format: yyyy-MM-dd
domain / dnssec	0 - n		Transmission of the key material is only needed if you sign the zone yourself. NOTE Please be aware that DNSSEC cannot be used with all of our nameservers. NOTE To remove the key material at the registry, use the value "false" in your request. <dnssec>>false</dnssec>
domain / dnssec / flags	1	enum	257 = KSK 256 = ZSK
domain / dnssec / protocol	1	int	Value: 3 (DNSSEC)
domain / dnssec / algorithm	1	enum	1=RSA/MD5 2=Diffie Hellman 3=DSA/SHA-1 4=Elliptische Kurven 5=RSA/SHA-1 7=RSASHA1-NSEC3-SHA1 8=RSA/SHA-256 10=RSA/SHA-512 12=ECC-GOST 13= ECDSAP256SHA256 14= ECDSAP384SHA384
domain / dnssec / publickey	1	Base64	Base64 coded public key
domain / extension / mode	0 - 1	enum	This key specifies how domain addons are treated when updating a domain. <ul style="list-style-type: none"> ▪ merge = all existing keys remain, if the value is different, it will be updated. New keys are added. ▪ replace = existing keys are deleted, the new ones are set. Default = merge

Key	Occurs	Format	Description
domain / extension / price_class	0 - 1	string	Required for premium domains. The price class can be inquired with the <i>Domain Premium Inquire (0165)</i> and the <i>Domain Premium Info (0164)</i> task.
domain / payable	1	date time	Billing date (Domain Renew) NOTE The billing date can be inquired with the DomainInquire task.
domain / period	0 - 1*	int	The number of years the registration period of the domain will be renewed for. (Domain Renew)
domain / remove_cancellation			Deprecated. Replaced by "domain / remove_cancelation"
domain / remove_cancelation	0 - 1	string	With this tag you can renew the domains and delete existing cancelations at once. Should an existing request for cancelation be deleted? <ul style="list-style-type: none"> • yes = the cancelation request will be deleted. • no = the cancelation request will not be deleted. Default: no When "no": If there is a current cancelation request the request will fail. (Domain Renew)
domain / use_auto_dnssec	0 - 1	enum	Activates or deactivates DNSSEC usage. Values: <ul style="list-style-type: none"> ▪ true = activates DNSSEC ▪ false = deactivates DNSSEC This keyword can only be used if the system completely manages the DNSSEC. This works for certain nameservers for example by using NodeSecure. To only manage the key material use the keyword DNSSEC.
show_handle_details	0 - 1	enum	Displays the domain contact details. <u>Available values:</u> ownerc, adminc, tech, zonec, all Separate severals contacts with comma.
overwrite_authinfo	0 - 1	enum	Overwriting AuthInfo when updating a domain. <u>Available values:</u> True: AuthInfo will be overwritten by a new one False: AuthInfo will not be overwritten by a new one. Default: true NOTE: Not available for registries which do not support a domain update task for updating an AuthInfo (e.g. Denic)

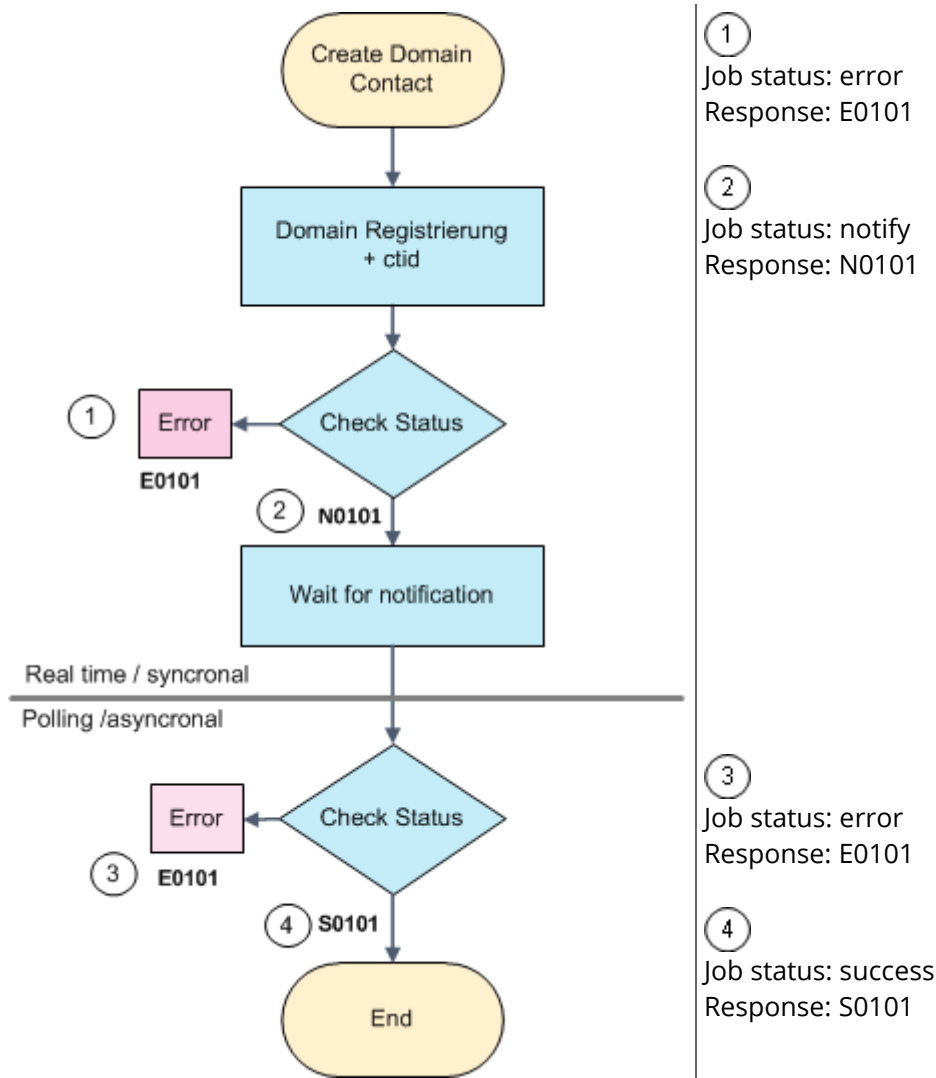
*) Depends on the registry.

4.2 Domain Create (0101)

With this task you can register new domains. All necessary domain contacts (handles) must be included in the request.

NOTE:

To register a premium domain you have to add the price_class key (<price_class>).



Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
</task>
```

```

<code>0101</code>
<domain>
  <name>example.com</name>
  <ctid>26515</ctid>
  <ownerc>9925345</ownerc>
  <adminc>9925345</adminc>
  <techc>9925345</techc>
  <zonec>9925345</zonec>
  <nserver>
    <name>ns1.example.com</name>
    <ip>172.16.100.1</ip>
    <ip6>2001:0DB8::1</ip6>
  </nserver>
  <nserver>
    <name>ns2.example.com</name>
    <ip>172.16.100.2</ip>
    <ip6>2001:0DB8::2</ip6>
  </nserver>
  <dnssec>
    <flags>257</flags>
    <protocol>3</protocol>
    <algorithm>5</algorithm>
    <publickey>key</publickey>
  </dnssec>
  <use_auto_dnssec>true</use_auto_dnssec>
  <confirm_order>1</confirm_order>
  <zone>
    <ip>172.16.100.3</ip>
    <mx></mx>
    <ns_action>complete</ns_action>
    <www_include>1</www_include>
  </zone>
  <use_trustee>1</use_trustee>
</domain>
<reply_to>customer@example.com</reply_to>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0101</code>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
      <text>Domain registration was started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>

```

4.3 Domain Create Plus (0101)

With this task you can register a new domain and create a new domain contact (handle) in a single task. You enter the data of a domain contact (handle) and the domain contact will then be generated automatically. If a contact with this data already exists, the existing contact will be used. In this case no new contact will be generated.

Within the “domain” element you can use all keywords which are allowed for the “normal” DomainCreate request.

In the respective domain contact element (ownerc, adminc, techc, zonec) you can use all keys, which are valid for creating a domain contact. Solely the “nic_ref” key cannot be used.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0101</code>
    <domain>
      <name>example.com</name>
      <ctid>x11111-1</ctid>
      <ownerc>
        <type>PERSON</type>
        <fname>Peter</fname>
        <lname>Doe</lname>
        <title></title>
        <organization>Example Company</organization>
        <address>Example Street 3</address>
        <pcode>12345</pcode>
        <city>Example City</city>
        <state></state>
        <country>US</country>
        <phone>+1-12345-127335</phone>
        <fax>+1-12345-12345</fax>
        <email>Doe@example.com</email>
        <sip></sip>
        <protection>B</protection>
        <nic_ref>info</nic_ref>
        <nic_ref>org</nic_ref>
        <remarks></remarks>
        <extension>
          <gender></gender>
          <idnumber></idnumber>
          <vatnumber></vatnumber>
          <trademark_name></trademark_name>
          <trademark_number></trademark_number>
          <trademark_country></trademark_country>
          <trademark_regdate></trademark_regdate>
          <trademark_appdate></trademark_appdate>
          <companynumber></companynumber>
          <birthday></birthday>
          <birthplace></birthplace>
          <birthcountry></birthcountry>
        </extension>
      </ownerc>
      <adminc>
        <type>PERSON</type>
        <fname>Hans</fname>
        <lname>Doe</lname>
        <title></title>
        <organization>Example Company</organization>
        <address>Example Street 3</address>
        <pcode>12745</pcode>
      </adminc>
    </domain>
  </task>
</request>
```



```

<city>Example City</city>
<state></state>
<country>US</country>
  <phone>+1-12345-166666</phone>
  <fax>+1-12345-127745</fax>
  <email>Doe@example.com</email>
  <sip></sip>
  <protection>B</protection>
  <nic_ref>info</nic_ref>
  <nic_ref>org</nic_ref>
  <remarks></remarks>
  <extension>
    <gender></gender>
    <idnumber></idnumber>
    <vatnumber></vatnumber>
    <trademark_name></trademark_name>
    <trademark_number></trademark_number>
    <trademark_country></trademark_country>
    <trademark_regdate></trademark_regdate>
    <trademark_appdate></trademark_appdate>
    <companynumber></companynumber>
    <birthday></birthday>
    <birthplace></birthplace>
    <birthcountry></birthcountry>
  </extension>
</adminc>
<techc>
  <type>PERSON</type>
  <fname>Hans</fname>
  <lname>Doe</lname>
  <title></title>
  <organization>Example Company</organization>
  <address>Example Street 3</address>
  <pcode>12345</pcode>
  <city>Example City</city>
  <state></state>
  <country>US</country>
  <phone>+1 -12345-166666</phone>
  <fax>+1-12345-127745</fax>
  <email>Doe@example.com</email>
  <sip></sip>
  <protection>B</protection>
  <nic_ref>info</nic_ref>
  <nic_ref>org</nic_ref>
  <remarks></remarks>
  <extension>
    <gender></gender>
    <idnumber></idnumber>
    <vatnumber></vatnumber>
    <trademark_name></trademark_name>
    <trademark_number></trademark_number>
    <trademark_country></trademark_country>
    <trademark_regdate></trademark_regdate>
    <trademark_appdate></trademark_appdate>
    <companynumber></companynumber>
    <birthday></birthday>
    <birthplace></birthplace>
    <birthcountry></birthcountry>
  </extension>
</techc>
<zonec>
  <type>PERSON</type>
  <fname>Hans</fname>
  <lname>Doe</lname>
  <title></title>
  <organization>Example Company</organization>
  <address>Example Street 3</address>
  <pcode>12345</pcode>

```

```

<city>MusterstadtExample City</city>
<state></state>
<country>US</country>
<phone>+1-12355-166666</phone>
<fax>+1-12345-133345</fax>
<email>Doe@example.com</email>
<sip></sip>
<protection>B</protection>
<nic_ref>info</nic_ref>
<nic_ref>org</nic_ref>
<remarks></remarks>
<extension>
  <gender></gender>
  <idnumber></idnumber>
  <vatnumber></vatnumber>
  <trademark_name></trademark_name>
  <trademark_number></trademark_number>
  <trademark_country></trademark_country>
  <trademark_regdate></trademark_regdate>
  <trademark_appdate></trademark_appdate>
  <companynumber></companynumber>
  <birthday></birthday>
  <birthplace></birthplace>
  <birthcountry></birthcountry>
</extension>
</zonec>
<nserver>
  <name>ns1.example.com</name>
</nserver>
<nserver>
  <name>ns2.example.com</name>
</nserver>
<!-- Mandatory for some TLDs.Für einige TLDs verpflichtend-->
<confirm_order>1</confirm_order>
<!-- Optional for some TLDsFür einige TLDs optional -->
<use_trustee>0</use_trustee>
<use_auto_dnssec>true</use_auto_dnssec>
</domain>
<reply_to>
  <email>MusterDoe@example.com</reply_to>
</task>
</request>

```

Example Response

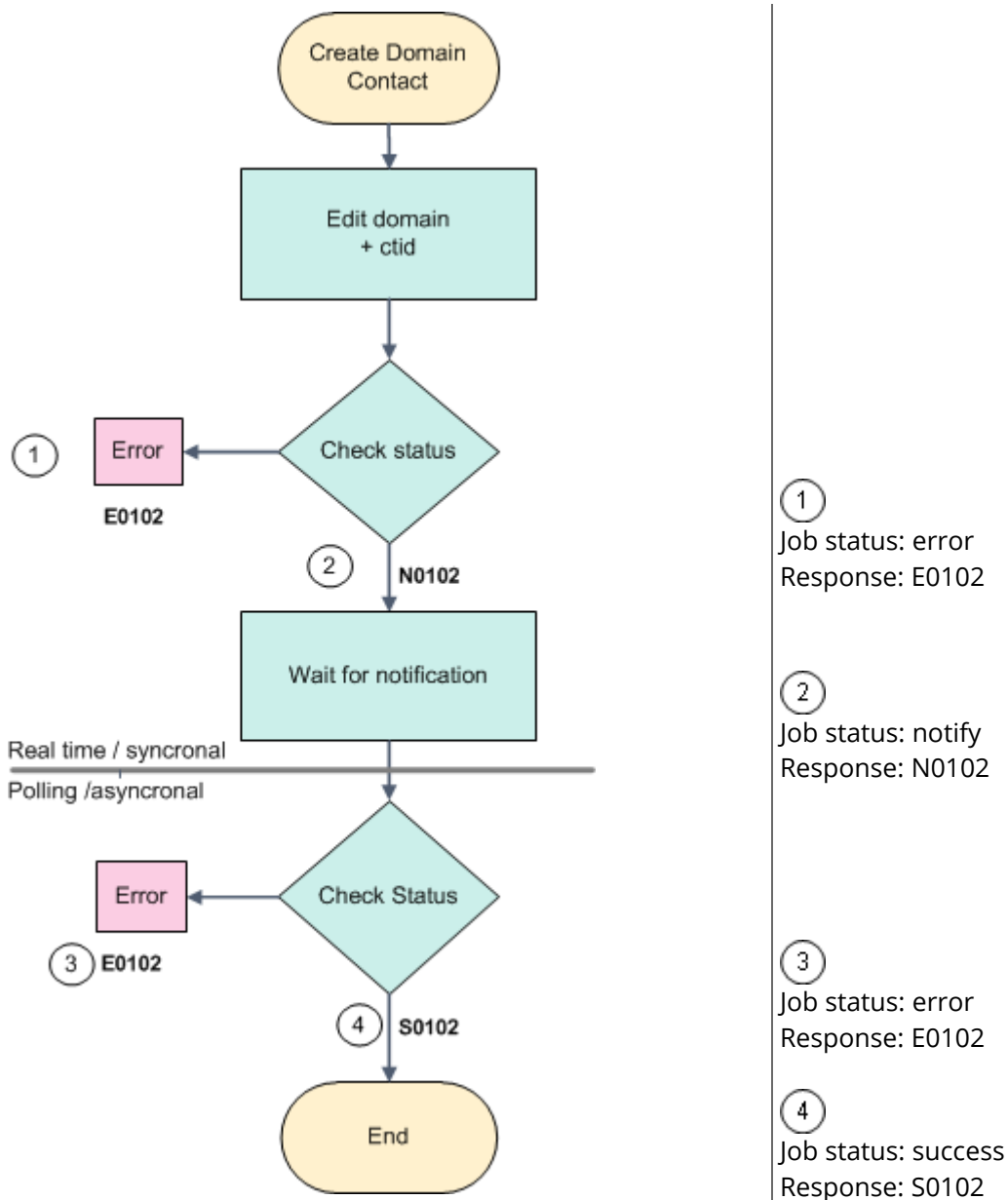
```

?xml version="1.0"encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0101001</code>
      <object>
        <type>owner-c</type>
        <value>9926617</value>
      </object>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
      <text>Domain registration was started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>

```

4.4 Domain Update (0102)

With this task you can update the domain data.



NOTE

The data of keys, which are not included in the request, will be loaded from the database.

NOTE: THE EFFECT OF AN EXISTING AUTHINFO1 ON AN OWNER CHANGE

If an owner change request is sent to the system, an existing AuthInfo1 will be deleted. If a cancelation request of the type preack exists, a new AuthInfo1 will be created automatically.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0102</code>
    <domain>
      <name>example.com</name>
      <ownerc>9925345</ownerc>
      <adminc>9925345</adminc>
      <techc>9925346</techc>
      <zonec>9925346</zonec>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
        <ip>172.16.100.1</ip>
        <ip6>2001:0DB8::1</ip6>
      </nserver> <overwrite_authinfo>>false</overwrite_authinfo>
      <dnssec>
        <flags>257</flags>
        <protocol>3</protocol>
        <algorithm>5</algorithm>
        <publickey>key...</publickey>
      </dnssec>
      <use_auto_dnssec>>true</use_auto_dnssec>
      <use_privacy>1</use_privacy>
    </domain>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response (synchronous request confirmation)

```
<?xml version="1.0" encoding="utf - 8" ?>
<response>
  <result>
    <status>
      <code>N0102</code>
      <text>Domain update has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

4.5 Domain Update Plus (0102)

With this task you can update a domain and create a new domain contact (handle) in a single task. You enter the data of a domain contact (handle) and the domain contact will then be generated automatically. If a contact with these data already exists, the existing contact will be used. In this case no new contact will be generated.

Within the "Domain" element all key words are possible which can be used with the "standard" DomainUpdate task too.

In the respective domain contact elements (ownerc, adminc, techc, zonec) all keys can be used which are also valid by updating a domain contact. Solely the "nic_ref" key cannot be used.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0102</code>
    <domain>
      <name>example.com</name>
      <ctid>x11111-1</ctid>
      <ownerc>
        <type>PERSON</type>
        <fname>Peter</fname>
        <lname>Doe</lname>
        <title></title>
        <organization>Example Company</organization>
        <address>Example Street 3</address>
        <pcode>12345</pcode>
        <city>Example City</city>
        <state></state>
        <country>US</country>
        <phone>+1-12345-127335</phone>
        <fax>+1-12345-12345</fax>
        <email>Doe@example.com</email>
        <sip></sip>
        <protection>B</protection>
        <nic_ref>info</nic_ref>
        <nic_ref>org</nic_ref>
        <remarks></remarks>
        <extension>
          <gender></gender>
          <idnumber></idnumber>
          <vatnumber></vatnumber>
          <trademark_name></trademark_name>
          <trademark_number></trademark_number>
          <trademark_country></trademark_country>
          <trademark_regdate></trademark_regdate>
          <trademark_appdate></trademark_appdate>
          <companynumber></companynumber>
          <birthday></birthday>
          <birthplace></birthplace>
          <birthcountry></birthcountry>
        </extension>
      </ownerc>
    </domain>
  </task>
</request>
```

```

</ownerc>
<adminc>
  <type>PERSON</type>
  <fname>Hans</fname>
  <lname>Doe</lname>
  <title></title>
  <organization>Example Company</organization>
  <address>Example Street 3</address>
  <pcode>12745</pcode>
  <city>Example City</city>
  <state></state>
  <country>US</country>
  <phone>+1-12345-166666</phone>
  <fax>+1-12345-127745</fax>
  <email>Doe@example.com</email>
  <sip></sip>
  <protection>B</protection>
  <nic_ref>info</nic_ref>
  <nic_ref>org</nic_ref>
  <remarks></remarks>
  <extension>
    <gender></gender>
    <idnumber></idnumber>
    <vatnumber></vatnumber>
    <trademark_name></trademark_name>
    <trademark_number></trademark_number>
    <trademark_country></trademark_country>
    <trademark_regdate></trademark_regdate>
    <trademark_appdate></trademark_appdate>
    <companynumber></companynumber>
    <birthday></birthday>
    <birthplace></birthplace>
    <birthcountry></birthcountry>
  </extension>
</adminc>
<techc>
  <type>PERSON</type>
  <fname>Hans</fname>
  <lname>Doe</lname>
  <title></title>
  <organization>Example Company</organization>
  <address>Example Street 3</address>
  <pcode>12345</pcode>
  <city>Example City</city>
  <state></state>
  <country>US</country>
  <phone>+1 -12345-166666</phone>
  <fax>+1-12345-127745</fax>
  <email>Doe@example.com</email>
  <sip></sip>
  <protection>B</protection>
  <nic_ref>info</nic_ref>
  <nic_ref>org</nic_ref>
  <remarks></remarks>
  <extension>
    <gender></gender>
    <idnumber></idnumber>
    <vatnumber></vatnumber>
    <trademark_name></trademark_name>
    <trademark_number></trademark_number>
    <trademark_country></trademark_country>
    <trademark_regdate></trademark_regdate>
    <trademark_appdate></trademark_appdate>
    <companynumber></companynumber>
    <birthday></birthday>
    <birthplace></birthplace>
    <birthcountry></birthcountry>
  </extension>

```

```

</techc>
<zonec>
  <type>PERSON</type>
  <fname>Hans</fname>
  <lname>Doe</lname>
  <title></title>
  <organization>Example Company</organization>
  <address>Example Street 3</address>
  <pcode>12345</pcode>
  <city>MusterstadtExample City</city>
  <state></state>
  <country>US</country>
  <phone>+1-12355-166666</phone>
  <fax>+1-12345-133345</fax>
  <email>Doe@example.com</email>
  <sip></sip>
  <protection>B</protection>
  <nic_ref>info</nic_ref>
  <nic_ref>org</nic_ref>
  <remarks></remarks>
  <extension>
    <gender></gender>
    <idnumber></idnumber>
    <vatnumber></vatnumber>
    <trademark_name></trademark_name>
    <trademark_number></trademark_number>
    <trademark_country></trademark_country>
    <trademark_regdate></trademark_regdate>
    <trademark_appdate></trademark_appdate>
    <companynumber></companynumber>
    <birthday></birthday>
    <birthplace></birthplace>
    <birthcountry></birthcountry>
  </extension>
</zonec>
<nserver>
  <name>ns1.example.com</name>
</nserver>
<nserver>
  <name>ns2.example.com</name>
</nserver>
<!-- Mandatory for some TLDs.Für einige TLDs verpflichtend-->
<confirm_order>1</confirm_order>
<!-- Optional for some TLDsFür einige TLDs optional -->
<use_trustee>0</use_trustee>
<use_auto_dnssec>true</use_auto_dnssec>
</domain>
<reply_to>
  <email>MusterDoe@example.com</email>
</reply_to>
</task>
</request>

```

Example Response

```

<?xml version="1.0"encoding="utf-8" ?>
<response>
  <result>
    <msg>
      <code>MSG01016</code>
      <status>success</status>
      <text>The missing data were replaced from the stock database.</text>
      <type>notice</type>
    </msg>
  </result>
  <status>
    <code>N0102001</code>
    <object>

```

```

        <type>owner-c</type>
        <value>1234567</value>
    </object>
</object>
    <type>domain</type>
    <value>example.com</value>
</object>
<text>Domain updates have been started successfully.</text>
<type>notify</type>
</status>
</result>
</response>

```

Example Request: Changing the OwnerC for a TLD for which the ownerchange is associated with costs. (here .eu)

```

<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>0102</code>
    <domain>
      <name>example9876.eu</name>
      <ownerc>
        <!--These contact data are not those of the OwnerC.-->
        <type>PERSON</type>
        <fname>Paul</fname>
        <lname>Doe</lname>
        <title></title>
        <organization></organization>
        <address>Example Street 3</address>
        <pcode>12367</pcode>
        <city>Example City</city>
        <state></state>
        <country>US</country>
        <phone>+1-12345-123456</phone>
        <fax>+1-12345-123457</fax>
        <email>Doe@example.com</email>
        <protection>B</protection>
        <remarks></remarks>
        <extension>
          <birthday>1970-01-01</birthday>
        </extension>
      </ownerc>
      <adminc>9910825</adminc>
      <techc>9910825</techc>
      <zonec>9910825</zonec>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
    </domain>
    <reply_to>customer@example.com</reply_to>
  </task>

```


Example Response

```
<?xml version="1.0"encoding="utf-8" ?>
<response>
  <result>
    <msg>
      <code>MSG01044</code>
      <status>success</status>
      <text>This TLD does not support an owner change.</text>
      <type>notice</type>
    </msg>
    <status>
      <code>N0102</code>
      <text>Domain update has been started successfully.</text>
      <text>D</text>
    </status>
  </result>
</response>
```

Example Request

```
<?xml version="1.0"encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>0102</code>
    <domain>
      <name>example.com</name>
      <ownerc>
        <type>PERSON</type>
        <fname>Max</fname>
        <lname>Customer</lname>
        <title></title>
        <organization></organization>
        <address>Hauptstr. 1</address>
        <pcode>12345</pcode>
        <city>Example City</city>
        <state></state>
        <country>US</country>
        <phone>+1-123-123</phone>
        <fax>+49+1-123-123</fax>
        <email>customer@example.com</email>
        <protection>B</protection>
        <remarks></remarks>
        <extension>
          <birthday>1970-01-01</birthday>
        </extension>
      </ownerc>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
    </domain>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response N0102001

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <msg>
      <code>MSG01016</code>
      <status>success</status>
      <text>The missing data were replaced from the stock database.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

4.6 Domain Inquire Tasks (0105)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
adminc	Domain administrator	yes			yes	yes	yes
authinfo	AuthInfo	yes			yes	yes	yes
autorenew	Usage of autoRenew.	yes			yes	yes	yes
cancelation	Cancelation type (DELETE, PREACK, TRANSIT)	yes			yes	yes	yes
comment	Domain comment	yes			yes	yes	yes
created	Date of registration	yes		yes		yes	yes
dnssec	DNSSEC. The Domain Info Task returns the complete object including sub items. The Domain List Task in contrast only returns the higher-level object.				yes		yes
dnssec / algorithm	DNSSEC algorithm	yes					
dnssec / flags	256 ZSK Zone Signing Key) or 257 KSK (Key Signing Key)	yes					
dnssec / protocol	Value always 3, because for DNSSEC context 3 is used.	yes					

Value	Description	Info	Info <key>	List	List <key>	order	where
dnssec / publickey	Base 64 coded public key	yes					
domainsafe	Usage of DomainSafe.	yes			yes		yes
expire	End of registration period	yes			yes	yes	yes
ip	Main IP address of the domain	yes					
name	Domain name	yes		yes		yes	yes
ns_group	Used premium nameserver group	yes			yes	yes	yes
nserver	Nameserver data				yes	yes	yes
nserver / name	Name of domain nameservers	yes					
nserver / ip	IPv4 nameserver address	yes					
nserver / ip6	IPv6 nameserver address	yes					
owner	Object user	yes		yes		yes	yes
owner / context	Owner of the domain object	yes		yes			
owner / user	Username of the object owner	yes		yes			
ownerc	Owner contact ID or the complete contact block	yes		yes		yes	yes
parking	Parking provider	yes			yes	yes	yes
payable	payable date	yes			yes	yes	yes
price_class	Price class. Only for premium domains	yes			yes	yes	yes
period	Registration period	yes			yes	yes	yes
registrar_status	The domain status at the registrar (Hold, Lock, Hold-Lock, Active, Auto-Lock)	yes			yes	yes	yes
registrar_status_r eason	Reason for the registrar status, currently only OCVAL	yes					
registry_status	Status at the registry (Hold, Lock, Hold- Lock, Active, Auto-Lock)	yes			yes	yes	yes
status	Status system message	yes			yes	yes	yes
status / code	System message code	yes					
status / text	System message text	yes					
status / type	Systemmessage type (success, error, notification)	yes					
techc	Technical administrator	yes			yes	yes	yes
updater	The user who made the last update.	yes			yes	yes	yes
updater /user	Name of the user who made the last update.	yes					
updater /context	Context of the user who made the last update.	yes					

Value	Description	Info	Info <key>	List	List <key>	order	where
updated	Last updated	yes		yes		yes	yes
use_privacy	Usage of WHOIS PRIVACY SERVICE.	yes			yes	yes	yes
use_trustee	Usage of Trustee service .	yes			yes	yes	yes
zonec	Zone administrator	yes			yes	yes	yes

4.6.1 Domain Info (0105)

With this task you can inquire domain data.

NOTE

For domain inquire there is an additional key `show_handle_details`. With this key it is possible to show the domain contact data. Allowed values are `ownerc`, `adminc`, `techc`, `zonec`, `all`. Several values can be used at once. Separate them with comma.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0105</code>
    <domain>
      <name>example.com</name>
    </domain>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <domain>
        <name>example.com</name>
        <expire>2009-08-28 14:45:19</expire>
        <ownerc>9926870</ownerc>
        <adminc>9910650</adminc>
        <techc>9910650</techc>
        <zonec>9910650</zonec>
        <registry_status>NONE</registry_status>
        <nserver>
          <name>ns1.example.com</name>
        </nserver>
        <nserver>
          <name>ns2.example.com</name>
        </nserver>
        <period>1</period>
```

```

    <remarks/>
    <status>SUCCESS</status>
    <use_trustee>>false</use_trustee>
    <use_privacy>>false</use_privacy>
    <domainsafe>>true</domainsafe>
    <autorenew>>true</autorenew>
    <extension/>
    <owner>
      <user>customer</user>
      <context>1018945</context>
    </owner>
    <updater>
      <user>customer</user>
      <context>1018945</context>
    </updater>
    <created>2009-08-28 14:45:17</created>
    <updated>2014-01-08 18:05:59</updated>
  </domain>
</data>
<status>
  <code>S0105</code>
  <text>Domain data has been inquired successfully.</text>
  <type>success</type>
  <object>
    <type>domain</type>
    <value>example.com</value>
  </object>
</status>
</result>
<ctid>customer-transaction-id</ctid>
<stid>20160204-appl-test-98130</stid>
</response>

```

Example Request mit Domainkontakt-Details

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0105</code>
    <domain>
      <name>example.com</name>
    </domain>
    <show_handle_details>ownerc</show_handle_details>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <domain>
        <ownerc>
          <address>abc</address>
          <alias>customer</alias>
          <city>city</city>
          <country>DEUS</country>
          <created>2008-04-08 12:58:21</created>
          <email>customer@example.com</email>
          <fax>+1-000-0000</fax>
          <fname>customer1</fname>

```

```

    <id>1234567</id>
    <lname>d</lname>
    <nic_ref>
      <name>DENIC--999999</name>
      <nic>de</nic>
      <status>success</status>
    </nic_ref>
    <nic_ref></nic_ref>
    <organization></organization>
    <owner>
      <context></context>
      <user>customer</user>
    </owner>
    <pcode>12345</pcode>
    <phone>+1-000-0000</phone>
    <protection>B</protection>
    <state>US</state>
    <type>PERSON</type>
  </ownerc>
  <name>example.com</name>
  <expire>2009-08-28 14:45:19</expire>
  <ownerc>9926870</ownerc>
  <adminc>9910650</adminc>
  <techc>9910650</techc>
  <zonec>9910650</zonec>
  <registry_status>NONE</registry_status>
  <nserver>
    <name>ns1.example.com</name>
  </nserver>
  <nserver>
    <name>ns2.example.com</name>
  </nserver>
  <period>1</period>
  <remarks/>
  <status>SUCCESS</status>
  <use_trustee>>false</use_trustee>
  <use_privacy>>false</use_privacy>
  <domainsafe>>true</domainsafe>
  <dnssec>>true</dnssec>
  <autorenew>>true</autorenew>
  <extension/>
  <owner>
    <user>customer</user>
    <context>1018945</context>
  </owner>
  <updater>
    <user>customer</user>
    <context>1018945</context>
  </updater>
  <created>2009-08-28 14:45:17</created>
  <updated>2014-01-08 18:05:59</updated>
</domain>
</data>
<status>
  <code>S0105</code>
  <text>Domain data has been inquired successfully.</text>
  <type>success</type>
  <object>
    <type>domain</type>
    <value>example.com</value>
  </object>
</status>
</result>
<ctid>customer-transaction-id</ctid>
<stid>20160204-appl-test-98130</stid>
</response>

```

4.6.2 Domain List (0105)

With the list inquire you receive certain data of the desired number of domains. Supplementary data can be returned in addition to the standard values by using certain values as a key term.

Examples for XML Requests and Answers

Example Request(list inquire)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0105</code>
    <view>
      <offset>0</offset>
      <limit>3</limit>
      <children></children>
    </view>
    <key>created</key>
    <where>
      <key>name</key>
      <operator>like</operator>
      <value>e*</value>
    </where>
    <key>domainsafe</key>
    <key>dnssec</key>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>13</summary>
      <domain>
        <name>example1695.com</name>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <created>2008-10-16 15:10:35</created>
        <updated>2015-12-05 15:10:49</updated>
        <domainsafe>true</domainsafe>
        <dnssec>true</dnssec>
      </domain>
      <domain>
        <name>example1234.se</name>
        <registry_status>NONE</registry_status>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <created>2009-05-21 20:45:33</created>
        <updated>2015-05-22 01:30:01</updated>
        <domainsafe>true</domainsafe>
        <dnssec>false</dnssec>
      </domain>
      <domain>
        <name>example.de</name>
```

```

        <registry_status>NONE</registry_status>
        <owner>
            <user>customer</user>
            <context></context>
        </owner>
        <created>2009-08-28 14:45:17</created>
        <updated>2014-01-08 18:05:59</updated>
        <domainsafe>>false</domainsafe>
        <dnssec>>false</dnssec>
    </domain>
</data>
<status>
    <code>S0105</code>
    <text>Domain data has been inquired successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20160215-app1-test-277</stid>
</response>

```

4.7 Domain Delete (0103)

With his task you can delete a request.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0103</code>
    <domain>
      <name>example.de</name>
      <transit>1</transit>
      <disconnect>0</disconnect>
      <execdate>2010-02-23</execdate>
    </domain>
  </task>
</request>

```

Example Response

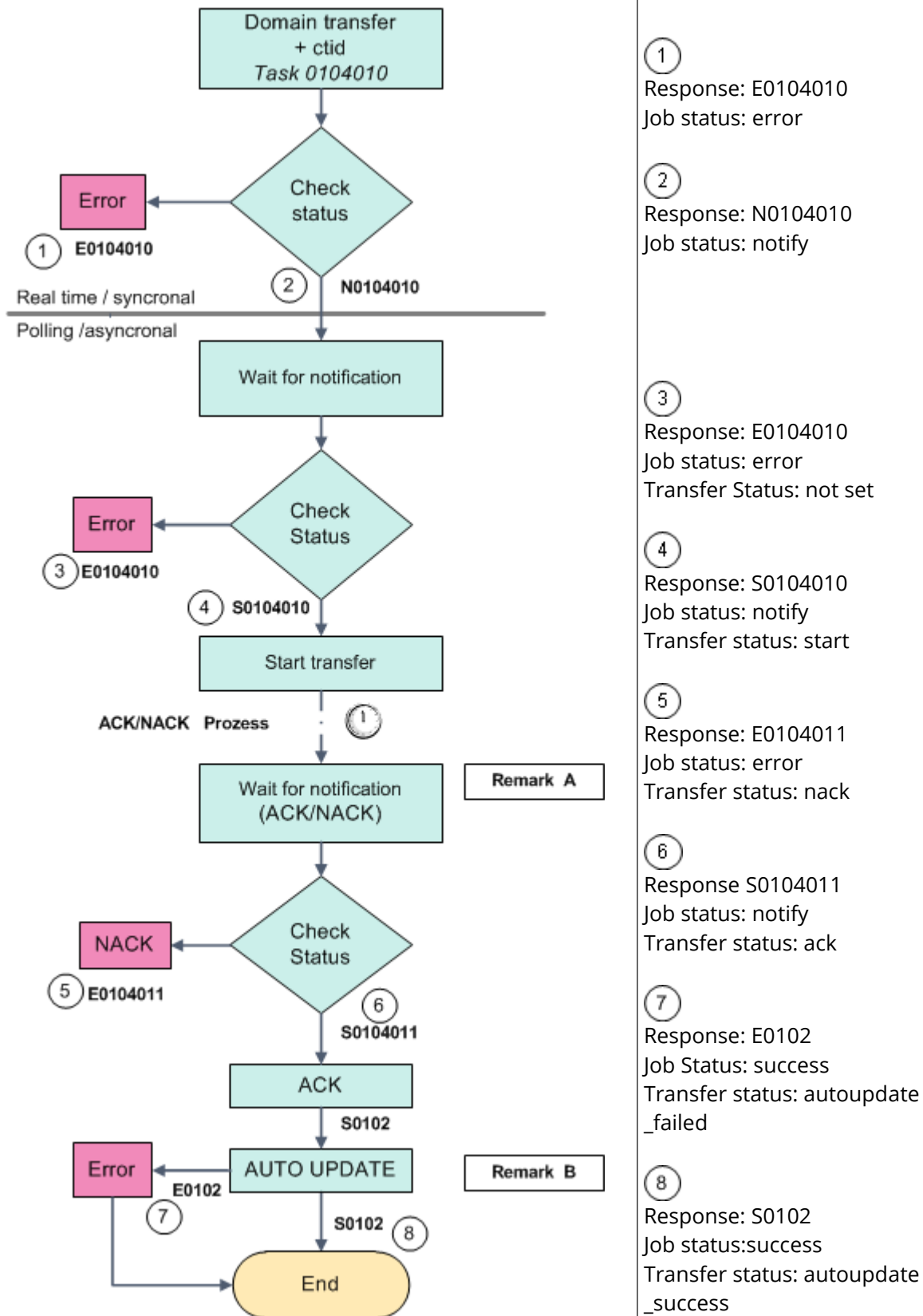
```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <cancelation>
        <created>2009-05-13 09:50:48</created>
        <disconnect>0</disconnect>
        <domain>example.de</domain>
        <execdate>2010-02-23 00:00:00</execdate>
        <type>TRANSIT</type>
      </cancelation>
    </data>
    <status>
      <code>N0103</code>
      <text>Domain deletion has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>

```


4.8 Domain Ownerchange (0104010)

With this task you can change the OwnerC domain contact.



- ① Response: E0104010
Job status: error
- ② Response: N0104010
Job status: notify
- ③ Response: E0104010
Job status: error
Transfer Status: not set
- ④ Response: S0104010
Job status: notify
Transfer status: start
- ⑤ Response: E0104011
Job status: error
Transfer status: nack
- ⑥ Response S0104011
Job status: notify
Transfer status: ack
- ⑦ Response: E0102
Job Status: success
Transfer status: autoupdate
_failed
- ⑧ Response: S0102
Job status:success
Transfer status: autoupdate
_success

Remarks

A Wait for notification (ACK/NACK)

You can ignore IRTP emails. They serve only for information purposes.

B Auto Update

If an AUTOUPDATE was carried out after the transfer, the tag „autoupdate: true“ is in the system block.

The same ctid is returned for the keyword “transaction” as with the transfer start and ACK.

NOTE

The *OwnerChange* task can only be used for some TLDs. You can use the *DomainTransfer* task to start a domain transfer in combination with an *OwnerChange* task. Some registries call this combination of tasks a trade.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0104010</code>
    <domain>
      <name>example.eu</name>
      <ownerc>9925345</ownerc>
      <techc>125085</techc>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
      <zone>
        <ns_action>hidden</ns_action>
      </zone>
    </domain>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example ResponseN0104010

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0104010</code>
      <text>Owner change has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

4.9 Domain Renew (0101003)

With this task you can renew the lifetime of a domain.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0101003</code>
    <domain>
      <name>example.com</name>
      <payable>2010-05-16</payable>
      <period>2</period>
      <remove_cancellation>yes</remove_cancellation>
    </domain>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <!--The ?msg? element will only sent back, if a cancellation had been existent.--> <msg>
      <code>MSG01046</code>
      <status>success</status>
      <text>The existing cancellation request was removed.</text>
      <type>notice</type>
    </msg>
    <status>
      <code>N0101003</code>
      <object>
        <type>domain</type>
        <value>example1695.com</value>
      </object>
      <text>Domain renew has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

4.10 Domain Premium Inquire (0165)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
created	Creation date of the preregistration	yes		yes		yes	yes
id	Internal ID of the order	yes		yes		yes	yes
name	Domain name	yes		yes		yes	yes
owner	Owner of the domain object	yes		yes		yes	yes
owner / context	Context of the object owner	yes		yes		yes	yes
owner / user	User name of the object owner	yes		yes		yes	yes
price_class	Price class of the premium domain	yes		yes		yes	yes
provider	Provider der TLD	yes		yes		yes	yes
status	Domain status	yes		yes		yes	yes
updater	The user who made the last update.	yes					
updater /user	Name of the user who made the last update.	yes					
updater /context	Context of the user who made the last update.	yes					
updated	Date of the last update of the preregistration	yes		yes		yes	yes

4.11 Domain Premium Info (0164)

With this task you can inquire information about the availability of the domain and its price class.

Example Request

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0164</code>
    <domain_premium>
      <name>premiumexample.com</name>
    </domain_premium>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?> <response>
  <result>
    <data>
      <domain_premium>
        <name>premiumexample.com</name>
        <price_class>1</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updater>
          <user>customer</user>
          <context></context>
        </updater>
        <id>1</id>
        <updated>2014-01-15 21:03:43</updated>
      </domain_premium>
    </data>
    <status>
      <code>S0164</code>
      <type>success</type>
      <object>
        <type>domain_premium</type>
        <value>premiumexample.com</value>
      </object>
    </status>
  </result>
</response>
```

4.12 Domain Premium Info (0164)

With this task you can inquire information about the availability of the domain and its price class.

Example Request

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0164</code>
    <domain_premium>
      <name>premiument.com</name>
    </domain_premium>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?> <response>
  <result>
    <data>
      <domain_premium>
        <name>premiument.com</name>
        <price_class>1</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updater>
          <user>customer</user>
          <context></context>
        </updater>
        <id>1</id>
        <updated>2014-01-15 21:03:43</updated>
      </domain_premium>
    </data>
    <status>
      <code>S0164</code>
      <type>success</type>
      <object>
        <type>domain_premium</type>
        <value>premiument.com</value>
      </object>
    </status>
  </result>
</response>
```

4.13 Domain Premium List (0165)

With this task you can inquire the domain data of some or all premium domains.

HINWEIS

Zum Registrieren einer Premium-Domain müssen Sie den den Auftragstyp *DomainCreate* verwenden und die Preisklasse (<price_class>) angeben.

NOTE:

To register a premium domain you have to use the *DomainCreate* task and add the price_class key.

Examples for XML Requests and Answers

Example Request (list inquire)

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0165</code>
    <view>
      <children>1</children>
      <offset>0</offset>
      <limit>3</limit>
    </view>
    <where>
      <and>
        <or>
          <key>name</key>
          <operator>like</operator>
          <value>e*</value>
        </or>
        <or>
          <key>name</key>
          <operator>like</operator>
          <value>s*</value>
        </or>
      </and>
    </where>
    <order>
      <key>name</key>
      <mode>asc</mode>
    </order>
    <key>status</key>
    <key>created</key>
    <key>owner</key>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>231962</summary>
      <domain_premium>
        <name>e.accountant</name>
        <price_class/>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>support</user>
          <context>1</context>
        </owner>
        <id>1500188</id>
        <created>2015-06-11 11:59:26</created>
        <updated>2015-06-11 11:59:26</updated>
      </domain_premium>
      <domain_premium>
        <name>e.accountants</name>
        <price_class>DONUTS-A+</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>support</user>
          <context>1</context>
        </owner>
        <id>4099035</id>
        <created>2015-12-14 11:29:17</created>
        <updated>2016-02-14 14:33:32</updated>
      </domain_premium>
      <domain_premium>
        <name>e.actor</name>
        <price_class>RIGHTSIDE-D</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>support</user>
          <context>1</context>
        </owner>
        <id>5066741</id>
        <created>2015-12-15 11:42:07</created>
        <updated>2016-02-13 00:59:34</updated>
      </domain_premium>
    </data>
    <status>
      <code>S0165</code>
      <text>The premium domains were inquired successfully.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20160218-app3-105467</stid>
</response>
```

Example Request

Search for .bayern premium domains and display three.

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
```



```

<code>0165</code>
<view>
  <children>1</children>
  <offset>0</offset>
  <limit>3</limit>
</view>
<where>
  <and>
    <or>
      <key>tld</key>
      <operator>eq</operator>
      <value>bayern</value>
    </or>
  </and>
</where>
<order>
  <key>name</key>
  <mode>asc</mode>
</order>
<key>status</key>
<key>created</key>
<key>owner</key>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>72514</summary>
      <domain_premium>
        <name>000.bayern</name>
        <price_class>BAYERN-TIER1</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>support</user>
          <context></context>
        </owner>
        <id>835879</id>
        <created>2015-01-08 14:13:01</created>
        <updated>2015-02-01 16:25:23</updated>
      </domain_premium>
      <domain_premium>
        <name>001.bayern</name>
        <price_class>BAYERN-TIER1</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>support</user>
          <context></context>
        </owner>
        <id>835880</id>
        <created>2015-01-08 14:13:01</created>
        <updated>2015-02-01 16:25:23</updated>
      </domain_premium>
      <domain_premium>
        <name>002.bayern</name>
        <price_class>BAYERN-TIER1</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>support</user>
          <context></context>
        </owner>

```

```

        <id>835881</id>
        <created>2015-01-08 14:13:01</created>
        <updated>2015-02-01 16:25:27</updated>
    </domain_premium>
</data>
<status>
    <code>S0165</code>
    <text>The premium domains were inquired successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20150202-app1-13690</stid>
</response>

```

Example Request

Searching all premium domains of the price class Donats-BBBB.

```

<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0165</code>
    <view>
      <children>1</children>
      <offset>0</offset>
      <limit>3</limit>
    </view>
    <where>
      <and>
        <key>price_class</key>
        <operator>eq</operator>
        <value>Donuts-BBBB</value>
      </and>
    </where>
    <order>
      <key>name</key>
      <mode>asc</mode>
    </order>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>21061</summary>
      <domain_premium>
        <name>000.accountants</name>
        <price_class>DONUTS-BBBB</price_class>
        <status>FREE</status>
        <provider>registry</provider>
        <owner>
          <user>support</user>
          <context></context>
        </owner>
        <id>817298</id>
        <created>2015-01-26 13:44:14</created>
        <updated>2015-01-30 04:15:03</updated>
      </domain_premium>
      <domain_premium>

```

```
<name>000.engineering</name>
<price_class>DONUTS-BBBB</price_class>
<status>FREE</status>
<provider>registry</provider>
<owner>
  <user>support</user>
  <context></context>
</owner>
<id>552436</id>
<created>2014-08-27 15:45:34</created>
<updated>2015-01-31 20:56:51</updated>
</domain_premium>
<domain_premium>
  <name>000.limited</name>
  <price_class>DONUTS-BBBB</price_class>
  <status>FREE</status>
  <provider>registry</provider>
  <owner>
    <user>support</user>
    <context></context>
  </owner>
  <id>624113</id>
  <created>2014-08-27 15:46:21</created>
  <updated>2015-02-01 13:37:55</updated>
</domain_premium>
</data>
<status>
  <code>S0165</code>
  <text>The premium domains were inquired.</text>
  <type>success</type>
</status>
</result>
<stid>20150202-app2-dev-2661</stid>
</response>
```

5 Preregistration

5.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	Domain Prereg Create 0110001 Domain Prereg Create Plus 0110001 Domain Prereg Confirm 0110007 Domain Prereg Update 0110002 Domain Prereg Delete 0110003 Domain Prereg Info 0110004 Domain Prereg List 0110005 Prereg Config List 0110006
domain _prereg	1 - n		
domain _prereg / name	1	string	Domain name
domain _prereg / phase	1	enum	The valid values depend on the TLD. <i>Examples</i> <ul style="list-style-type: none"> • Sunrise2a • Sunrise2b • Landrush • Live
domain _prereg / nserver	5		Nameserver data
domain _prereg / nserver / name	1	domain	Nameserver name. Glue records are not allowed.
domain _prereg / tmch_mark	0 - 1		
domain _prereg / tmch_mark / reference	1	string	Reference number of the TMCH mark
domain _prereg / ownerc	1	contact-id*	Domain contact of the domain owner
domain _prereg / adminc	1	contact-id*	Domain contact of the domain administrator
domain _prereg / techc	1	contact-id*	Domain contact of the technical contact
extension / ipr_name	1	string	(ipr = intellectuall property rights) Name of the mark or right
extension / ipr_number	1	string	Number of the mark or right
extension / ipr_cc_locality	1	enum	Country code of the country in which the mark is registered (see WIPO list in the addendum)
extension / ipr_app_date	1	datum	Date on which the mark was applied for.
extension / ipr_reg_date	1	datum	Date on which the mark was registered
extension / ipr_app_capacity	1	enum	Funktion of the applicant. <u>Available values:</u> <ul style="list-style-type: none"> • owner • licensee • assignee

* contact id or contact data in the plus mode

Description of the statuses of the preregistration process

Status	Description
accept	The domain was registered successfully. For FCFS (first come first served) and APPLICATION).
auto_cancel	The domain has already been assigned.
decline	The registry denied the registration. Is to be seen similar to a "invalid_name".
failed	By first-come, first, served (FCFS) the domain is already assigned, otherwise a miscellaneous error has occurred.
invalid_name	The domain name is not available.
open	The customer did not yet confirm the preregistration. These domains will not be processed.
pending	The customer has confirmed the domain preregistration.
pending_sent	The application was sent to the registry.
For TMCH domains	
tmch_claim	A trademark exists for this domain. The prospective domain owner must additionally confirm that he wants this domain.
tmch_claim_pending	TMCH is in progress. Waiting for tmch_claim_confirmed oder tmch_claim_expired.
tmch_claim_confirmed	The prospective domain owner has confirmed that he wants to register this domain.
tmch_claim_expired	The prospective domain owner has neither confirmed nor rejected his request within the confirmation time frame.
tmch_claim_rejected	The request was automatically cancelled because the TMCH Claims Notice was not confirmed within the allotted time of 15 days.

5.2 Domain Prereg Create (0110001)

With this task you can pre-register domains. The request has to include all required domain contacts. Please note that the preregistration phase of every new TLD is limited in length and scope respective the terms and conditions of the registry.

NOTE

Please bear in mind that the preregistration phases of each new TLD are timely limited by the regulations of the respective registry.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>user</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0110001</code>
    <domain_prereg>
      <name>example.web</name>
      <tmch_mark>
        <reference>0000061236-1</reference>
      </tmch_mark>
      <phase>live</phase>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
    </domain_prereg>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data/>
    <status>
      <code>S0110001</code>
      <text>The domain has been preregistered successfully.</text>
      <type>success</type>
      <object>
        <type>domain_prereg</type>
        <value>30902702-0c7c-4de6-8d4d-6b39dba5ddfc</value>
      </object>
    </status>
  </result>
  <stid>20131202-app1-dev-3335</stid>
</response>
```

5.3 Domain Prereg Create Plus (0110001)

With this task you can preregister a domain and create a new domain contact (handle) in a single task. You enter the data of a domain contact (handle) and the domain contact will then be generated automatically. If a contact with this data already exists, the existing contact will be used. In this case no new contact will be generated.

NOTE

Please bear in mind that the preregistration phases of each new TLD are timely limited by the regulations of the respective registry.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0110001</code>
    <domain_prereg>
      <name>example.app</name>
      <tmch_mark>
        <reference>100115-1</reference>
      </tmch_mark>
      <phase>live</phase>
      <ownerc>
        <type>PERSON</type>
        <fname>Joseph</fname>
        <lname>MusterDoe</lname>
        <address>MusterstrasseExample Street 123</address>
        <pcode>23445</pcode>
        <city>MusterstadtExample City</city>
        <country>DEUS</country>
        <state></state>
        <phone>+49+1-12345-127335+49-0-00</phone>
        <fax>+49+1-12345-127335+49-0-00</fax>
        <email>MusterDoe@example.com</email>
        <protection>A</protection>
        <title>Prof.</title>
        <organization>BeispielfirmaExample Company</organization>
        <extension>
          <vatnumber>VATNUMBER</vatnumber>
          <trademark_number>VATNUMBER</trademark_number>
        </extension>
      </ownerc>
      <adminc>
        <type>PERSON</type>
        <fname>Sue</fname>
        <lname>MusterDoe</lname>
        <address>MusterstrasseExample Street 123</address>
        <pcode>23445</pcode>
        <city>MusterstadtExample City</city>
        <country>DEUS</country>
        <state></state>
        <phone>+49+1-12345-127335</phone>
        <fax>+49+1-12345-127335</fax>
```

```

        <email>customer@example.com</email>
        <protection>A</protection>
        <title>Prof.</title>
        <organization>BeispielfirmaExample Company</organization>
        <extension>
            <vatnumber>VATNUMBER</vatnumber>
            <trademark_number>TRADEMARK_NUMBER</trademark_number>
        </extension>
    </adminc>
    <techc>21367745</techc>
    <zonec>21367745</zonec>
    <nserver>
        <name>ns1.example.com</name>
    </nserver>
    <nserver>
        <name>ns2.example.com</name>
    </nserver>
</domain_prereg>
<reply_to>user@example.com</reply_to>
<default>
    <phase>live</phase>
</default>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response
  <result>
    <data/>
    <msg>
      <text>Domain contact has been successfully created.</text>
      <code>S0301</code>
      <type>success</type>
    </msg>
    <msg>
      <text>This domain contact already exists and is used.
      A new contact was not created </text>.
      <code>MSG03003</code>
      <type>notice</type>
    </msg>
    <object>
      <type>handle</type>
      <value>21387944</value>
    </object>
    </msg>
    <msg>
      <text>Domain contact has been successfully created.</text>
      <code>S0301</code>
      <type>success</type>
    </msg>
    <msg>
      <text>This domain contact already exists and is used.
      A new contact was not created </text>.
      <code>MSG03003</code>
    </msg>
    <text>This domain contact already exists and is used.
    A new contact was not created </text>.
      <code>MSG03003</code>
      <type>notice</type>
    </msg>
    <object>
      <type>handle</type>

```



```

        <value>21387945</value>
      </object>
    </msg>
    <status>
      <code>S0110001</code>
      <text>The domain has been preregistered successfully.</text>
      <type>success</type>
      <object>
        <type>domain_prereg</type>
        <value>8f797b77-f4c4-4c2d-bffb-01b17bec4f49</value>
      </object>
    </status>
  </result>
  <stdid>20131202-app1-dev-3457</stdid>
</response>

```

5.4 Domain Prereg Confirm (0110007)

This function makes the order a binding registration.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>user</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0110007</code>
    <domain_prereg>
      <order_reference>1b3d6110-c4ac</order_reference>
    </domain_prereg>
    <reply_to>user@example.com</reply_to>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data />
    <status>
      <code>S0110007</code>
      <type>success</type>
      <object>
        <type>domain_prereg</type>
        <value>1c3d6112-c4ac</value>
      </object>
    </status>
  </result>
</response>

```

5.5 Domain Prereg Update (0110002)

With this task you can change the data of a preregistration quest. Please bear in mind that a preregistration can be edited only in the status "pending". To change the preregistration phase (e.g. from landrush to sunrise) you must use the key "phase_new".

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0110002</code>
    <domain_prereg>
      <order_reference>1c2d6112-c1ac</order_reference>
      <tmch_mark>
        <reference>0000061236-1</reference>
      </tmch_mark>
      <phase>live</phase>
      <phase_new>sunrise1</phase_new>
      <ownerc>12345678</ownerc>
      <adminc>12345678</adminc>
      <techc>12345678</techc>
      <zonec>12345678</zonec>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
    </domain_prereg>
    <reply_to>user@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf - 8" ?>
<response>
  <result>
    <data />
    <status>
      <code>S0110002</code>
      <type>success</type>
      <object>
        <type>domain_prereg</type>
        <value>1c3d6112-c4ac</value>
      </object>
    </status>
  </result>
</response>
```

5.6 Domain Prereg Delete (0110003)

With this task you can delete a preregistration. Please be aware that a preregistration can only be deleted in the status „open“.

NOTE:

You can delete a preregistration in two ways: either by submitting the domain name and the registration phase for which it was registered in or alternatively the reference number. The order reference number can be queried beforehand with the domain prereg inquire function 0110004 (single inquire).

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0110003</code>
    <domain_prereg>
      <name>example.com</name>
      <phase>sunrise2b</phase>
    </domain_prereg>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data/>
    <status>
      <code>S0110003</code>
      <type>success</type>
      <object>
        <type>domain_prereg</type>
        <value>1c3d6112-c4ac-4de4-9670-42c48e7456ac</value>
      </object>
    </status>
  </result>
</response>
```

5.7 Domain Prereg Inquire Tasks (0110004, 0110005)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
adminc	Domain administrator	yes		yes		yes	yes
confirmed	Confirmed. Values: True/False	yes		yes		yes	yes
created	Creation date of the preregistration	yes		yes		yes	yes
idn	Original writing of the domain name with umlaute	yes		yes			
name	Domain name	yes		yes		yes	yes
extension	Extensions concerning IPR (ipr = intellectuell property rights)	yes					
ns_action	Namesever mode	yes					
nserver	Domain nameservers	yes		yes		yes	yes
nserver / name	Name of domain nameservers						
order_reference	Reference number of the order	yes		yes		yes	yes
owner	Object owner	yes		yes		yes	yes
owner / context	Owner of the domain object	yes					
owner / user	User name of the object owner	yes		yes			
period	Registration period	yes					
prereg_config	Preregistration configuration			yes			
prereg_config / confirmation_start	Start of confirmation period			yes			
prereg_config / confirmation_end	End of confirmation period			yes			

Value	Description	Info	Info <key>	List	List <key>	order	where
prereg_config / period_start	Start of registration period			yes			
prereg_config / period_end	End of registration period			yes			
phase	Registration period	yes		yes		yes	yes
status	Status of the preregistration	yes		yes		yes	yes
techc	Technical administrator	yes		yes		yes	yes
tmch_mark	TMCH mark	yes					
tmch_mark /reference	reference number of the TMCH mark	yes					
updated	Lat updated	yes		yes		yes	yes
updater	User who made the last update.	yes					
updater /user	Name of the user who made the last update.	yes					
updater /context	Context of the user who made the last update.	yes					

5.8 Domain Prereg Info (0110004)

With this task you can inquire the data of the preregistration request including the status.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>user</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0110004</code>
    <domain_prereg>
      <name>example.web</name>
      <phase>live</phase>
    </domain_prereg>
    <reply_to>user@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <domain_prereg>
        <name>magic-chicken.author</name>
        <ownerc>11344248</ownerc>
      </domain_prereg>
    </data>
  </result>
</response>
```

```

    <adminc>11344248</adminc>
    <techc>13302751</techc>
    <zonec>13302751</zonec>
    <nserver>
      <name>ns1.example.com</name>
    </nserver>
    <nserver>
      <name>ns1.example.com</name>
    </nserver>
    <ns_action>NONE</ns_action>
    <period>0</period>
    <status>OPEN</status>
    <phase>live</phase>
    <extension/>
    <order_reference>48ff5985-fabf</order_reference>
    <use_trustee>0</use_trustee>
    <owner>
      <user>customer</user>
      <context></context>
    </owner>
    <updater>
      <user>customer2</user>
      <context></context>
    </updater>
    <created>2015-08-12 13:47:14</created>
    <updated>2015-08-12 13:47:14</updated>
  </domain_prereg>
</data>
<status>
  <code>S0110004</code>
  <text>Domain preregistration data was inquired successfully.</text>
  <type>success</type>
  <object>
    <type>domain_prereg</type>
    <value>example.web</value>
  </object>
</status>
</result>
<stid>20150812-appl-test-5807</stid>
</response>

```

5.9 Domain Prereg List (0110005)

With this task you can inquire your preregistrations.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>0110005</code>
    <view>
      <children>1</children>
      <offset>0</offset>
      <limit>10</limit>
    </view>
    <where>

```

```

        <and>
            <key>tld</key>
            <operator>like</operator>
            <value>web</value>
        </and>
    </where>
    <key>adminc</key>
    <key>nserver</key>
    <key>owner</key>
    <key>ownerc</key>
    <key>techc</key>
    <key>zonec</key>
    <key>phase</key>
    <order>
        <key>created</key>
        <mode>asc</mode>
    </order>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <summary>1</summary>
      <domain_prereg>
        <name>example1.web</name>
        <ownerc>199563</ownerc>
        <adminc>199563</adminc>
        <techc>199563</techc>
        <zonec>199563</zonec>
        <nserver>
          <name>ns1.example.com</name>
        </nserver>
        <nserver>
          <name>ns2.example.com</name>
        </nserver>
        <status>PENDING</status>
        <phase>sunrise1</phase>
        <order_reference>1c3d6112-c4ac</order_reference>
        <prereg_config>
          <confirmation_start>2013-01-24
            00:00:00</confirmation_start>
          <confirmation_end>2044-12-24
            00:00:00</confirmation_end>
          <period_start>2013-01-24 00:00:00</period_start>
          <period_end>2014-01-24 00:00:00</period_end>
        </prereg_config>
        <owner>
          <user>customer</user>
          <context>104</context>
        </owner>
        <updated>2013-10-09 12:36:58</updated>
      </domain_prereg>
    </data>
    <status>
      <code>S0110005</code>
      <type>success</type>
    </status>
  </result>
</response>

```

5.10 Prereg Config List (0110006)

With this task you can inquire the details of preregistration phases.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>user</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0110006</code>
    <view>
      <children>1</children>
      <offset>0</offset>
      <limit>10</limit>
    </view>
    <where>
      <and>
        <key>name</key>
        <operator>like</operator>
        <value>*</value>
      </and>
      <and>
        <key>phase</key>
        <operator>like</operator>
        <value>live*</value>
      </and>
    </where>
    <key>phase</key>
    <key>period_start</key>
    <order>
      <key>name</key>
      <mode>asc</mode>
    </order>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <summary>2</summary>
      <prereg_config>
        <name>Example1</name>
        <phase>live</phase>
        <confirmation_start>2013-09-24 08:23:39</confirmation_start>
        <confirmation_end>2014-09-24 08:23:42</confirmation_end>
        <period_start>2013-09-24 08:24:09</period_start>
        <period_end>2014-09-24 08:24:11</period_end>
        <created>2013-09-24 08:25:04</created>
        <updated>2013-09-24 08:24:05</updated>
      </prereg_config>
      <prereg_config>
        <name>Example2</name>
        <phase>liveOK</phase>
        <confirmation_start>2013-01-24 08:23:39</confirmation_start>
```



```
        <confirmation_end>2014-12-24 08:23:42</confirmation_end>
        <period_start>2013-01-24 08:24:09</period_start>
        <period_end>2014-01-24 08:24:11</period_end>
        <created>2013-09-24 08:25:04</created>
        <updated>2013-09-24 08:24:05</updated>
    </prereg_config>
</data>
<status>
    <code>S0110006</code>
    <type>success</type>
</status>
</result>
</response>
```

6 Domain Cancelation and Domain Restore

6.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	Domain Cancelation Create 0103101 Domain Cancelation Update 0103102 Domain Cancelation Delete 0103103 Domain Cancelation Info 0103104 Domain Cancelation History Info 0103124 Domain Cancelation History List 0103125 Domain Restore List 0105005 Domain Restore 0101005
cancelation	1 - n		
cancelation / domain	1	string	The domain name
cancelation / type	1	enum	Cancelation types. Values: <ul style="list-style-type: none"> • delete: cancels the domain • transit: gives the domain back to the registry. This is only possible for some TLDs (e.g. .de). • preack: releases the domain in advance for a domain transfer.
cancelation / execdate	0 - 1	string	Execution date. <u>Available values:</u> <ul style="list-style-type: none"> • yyyy-MM-dd • yyyy-MM-dd hh:mm:ss • expire: cancelation at expiration date • now: immediate cancelation
cancelation / disconnect	0 - 1*	enum	Here you can define if the zone will be disconnected or not. <ul style="list-style-type: none"> • 0= zone will not be disconnected • 1= zone will be disconnected Default= 0 Required for transit only.
cancelation / gaining registrar	0 - 1	string	Required for preack only. The registrar, the domain will be transferred to. Values: <ul style="list-style-type: none"> • accept_all = all registrars are possible • Enter a certain registrar (e.g. DENIC-4) Ask the registry for the ID of your registrar. In this case the domain can be transferred to this registrar only.
action	0 - 1	string	Restore type. Depends on the restore phase of the domain. <u>Available values:</u> <ul style="list-style-type: none"> ▪ RESTORE_ARGP (= Auto Renew Grace Period) will be restored free of charge.

Key	Occurs	Format	Description
			<ul style="list-style-type: none"> RESTORE_RGP (= Redemption Grace Period) are always associated with costs. Please see the current price list. RESTORE_RENEW. restores domains in the ARGP phase or and renewing a domain

6.2 Domain Cancellation Create (0103101)

With this task you can create a delayed cancellation of a domain. There are three possible forms of cancellation:

- Delete: deletion of the domain
- Transit: domain returns to the registry
- Preack: pre-acknowledged transfer

As long as the cancellation is in process, it is not possible to update or delete the cancellation request.

NOTE CONCERNING PREACK

If an AuthInfo1 already exists for the domain you want to cancel, it is not possible to start a cancellation of the type preack. You'll get the error message "Authinfo1 is already present." Delete AuthInfo1 first. Then you can start a cancellation of the type preack.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0103101</code>
    <cancellation>
      <domain>example.com</domain>
      <type>delete</type>
      <execdate>expire</execdate>
      <disconnect>1</disconnect>
      <gaining_registrar></gaining_registrar>
    </cancellation>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <cancellation>
```

```

        <created>2009-05-1311:02:42</created>
        <disconnect>0</disconnect>
        <domain>example.com</domain>
        <execdate>2009-12-03 12:14:27</execdate>
        <type>DELETE</type>
    </cancellation>
</data>
<status>
    <code>S0103101</code>
    <text>Domain cancellation has been started successfully.</text>
    <type>success</type>
</status>
</result>
</response>

```

6.3 Domain Cancellation Update (0103102)

With this task you can update an existing request for delayed cancellation. You can update the type (delete, transit, preack) and the date of execution.

NOTE: THE BEHAVIOUR OF AN EXISTING AUTHINFO1

Updating a cancellation of the type preack, an existing AuthInfo1 will be deleted regardless of the kind of changes. If the new request also is a cancellation of the type preack, a new AuthInfo1 will be created automatically.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0103102</code>
    <cancellation>
      <domain>example.com</domain>
      <type>delete</type>
      <execdate>2010-02-01</execdate>
      <disconnect>1</disconnect>
    </cancellation>
    <reply_to>customer@example.com</reply_to>
    <ctid>customer-transaction-id</ctid>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <cancellation>
        <disconnect>0</disconnect>
        <domain>example.com</domain>
        <execdate>2010-02-01 00:00:00</execdate>
        <type>delete</type>

```

```

      <updated>2009-05-13 11:25:58</updated>
    </cancelation>
  </data>
  <status>
    <code>S0103102</code>
    <text>Domain cancelation was updated successfully.</text>
    <type>success</type>
  </status>
</result>
</response>

```

6.4 Domain Cancelation Delete (0103103)

With this task you can delete an existing, not yet executed request for delayed cancelation.

NOTE

Some TLDs have fixed cancelation periods. They will be checked by the system when the cancelation delete request comes in. If the cancelation period was not kept, the domain will be renewed. A cancelation will not be possible until the end of the new registration period. For some TLDs only the execution date (execdate key) „expire“ is possible. If you have sent the request with the execution date “now” or a defined date, it will be changed to “expire”.

NOTE: THE BEHAVIOUR OF AN EXISTING AUTHINFO1

Deleting a cancelation of the type Preack, an existing AuthInfo1 will always be deleted.

The following table shows the concerned TLDs:

TLD	Cancelation period will be checked	Execution Date will be changed to “expire”
.at	yes	no
.es	yes	yes
.cz	yes	no
.se	yes	yes
.si	yes	yes

NOTE

For the majority of TLDs you can cancel a cancelation request until 12 hours before the defined cancelation date. To cancel the cancelation request at a later date, contact our support to be informed about the possibilities.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
  </auth>
</request>

```

```

</auth>
<task>
  <code>0103103</code>
  <cancelation>
    <domain>example.com</domain>
    <execdate>now</execdate>
  </cancelation>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0103103</code>
      <text>Cancelation has been deleted successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

6.5 Domain Cancelation Inquire Tasks (0103104)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
authinfo	AuthInfo. For the cancelation type PREACK only				yes		
changed	Date of last changes	yes			yes	yes	yes
created	Creation date of the cancelation	yes			yes	yes	yes
ctid	Client transaction id				yes	yes	yes
disconnect	Should the domain be disconnected after the cancelation? True/yes or false/no.				yes	yes	yes
domain	Domain name	yes			yes	yes	yes

Value	Description	Info	Info <key>	List	List <key>	order	where
execdate	Cancelation date	yes			yes	yes	yes
gaining_registrar	Future Registrar				yes	yes	yes
owner	Object owner	yes			yes	yes	yes
registry_when_date	The point of time when the request was carried out at the registry	yes			yes	yes	yes
reply_to	Email reply address				yes		
registry_status	Domain status at the registry				yes	yes	yes
spool_when	The point of time when the request was carried out by the system.				yes	yes	yes
type	Cancelation Type (DELETE, TRANSIT, PREACK)	yes			yes	yes	yes
updater	The user who made the last update	yes			yes	yes	yes

6.6 Domain Cancelation Info (0103104)

With this task you can inquire the data of a single current cancelation request.

Examples for XML Requests and Answers

Hinweis:

Mit diesem Auftragstyp können auch Massenaufträge (Bulk) ausgeführt werden.

Example Request(single inquire)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0103104</code>
    <cancelation>
      <domain>example.com</domain>
    </cancelation>
  </task>
</request>
```

Example Response

```
?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <cancelation>
        <execdate>expire</execdate>
        <changed>2015-12-19 21:58:01</changed>
        <domain>example.com</domain>
        <type>DELETE</type>
        <registry_when_date>2017-09-22 22:30:31</registry_when_date>
```

```

        <owner>
            <user>customer</user>
            <context>4</context>
        </owner>
        <updater>
            <user>customer</user>
            <context>4</context>
        </updater>
        <created>2015-12-19 21:58:01</created>
    </cancelation>
</data>
<status>
    <code>S0103104</code>
    <text>Die Kuendigungsauftraege wurden erfolgreich ermittelt.</text>
    <type>success</type>
    <object>
        <type>domain</type>
        <value>albris.co.uk</value>
    </object>
</status>
</result>
<stid>20160119-app3-59430</stid>
</response>

```

6.7 Domain Cancelation List (0103104)

With this task you can inquire a list of all current open requests for delayed cancelation, too.

Examples for XML Requests and Answers

Example Response(list inquire)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0103104</code>
    <view>
      <offset>0</offset>
      <limit>10</limit>
      <children>1</children>
    </view>
    <where>
      <key>domain</key>
      <operator>like</operator>
      <value>example*</value>
    </where>
    <key>disconnect</key>
    <key>execdate</key>
    <key>gaining_registrar</key>
  </task>
</request>

```


Example Response(list inquire)

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <summary>2</summary>
      <cancelation>
        <execdate>expire</execdate>
        <disconnect>1</disconnect>
        <domain>example.info</domain>
        <type>DELETE</type>
      </cancelation>
      <cancelation>
        <domain>example.com</domain>
        <type>TRANSIT</type>
      </cancelation>
    </data>
    <status>
      <type>success</type>
      <code>S0103104</code>
      <text>Cancelations have been inquired successfully.</text>
    </status>
  </result>
</response>
```

6.8 Domain Cancelation History Inquire Tasks (0103124, 0103125)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
changed	Date of the last update	yes			yes	yes	yes
created	Creation date of the request	yes			yes	yes	yes
ctid	The CTID of the request (Client Transaction ID)				yes	yes	yes
disconnect	Is the domain disconnected? Values: Yes/True und No/False				yes	yes	yes
domain	Domain name	yes			yes	yes	yes

Value	Description	Info	Info <key>	List	List <key>	order	where
execdate	Execution date of the cancelation	yes		yes		yes	yes
gaining_registrar	Future registrar				yes	yes	yes
log_id	Request id in the history	yes		yes		yes	yes
owner	Object owner	yes			yes	yes	yes
registry_when_date	The point of time when the request was carried out at the registry	yes			yes	yes	yes
spool_when	The point of time when the request was carried out by the system.				yes	yes	yes
type	Cancelation type (DELETE, PREACK, TRANSIT)	yes			yes	yes	yes
updater	The user who made the last update	yes			yes	yes	yes

6.9 Domain Cancelation History Info (0103124)

With this task you can inquire the data of an already processed cancelation request.

Examples for XML Requests and Answers

Example Request(single inquire)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0103124</code>
    <cancelation>
      <log_id>32539</log_id>
    </cancelation>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <cancelation>
        <execdate>expire</execdate>
        <changed>2014-06-03 14:46:17</changed>
        <domain>example.com</domain>
        <type>DELETE</type>
        <spool_when>2015-06-03 02:45:51</spool_when>
        <registry_when_date>2015-06-03 14:45:51</registry_when_date>
        <log_id>32539</log_id>
```

```

        <owner>
            <user>customer</user>
            <context></context>
        </owner>
        <updater>
            <user>customer</user>
            <context>1</context>
        </updater>
        <created>2014-06-03 14:46:17</created>
    </cancellation>
</data>
<status>
    <code>S0103124</code>
    <text>Domain cancellation request history has been inquired.</text>
    <type>success</type>
</status>
</result>
<stid>20140603-appl-dev-4644</stid>
</response>

```

6.10 Domain Cancellation History List (0103125)

With this task you can inquire a list all already processed cancellations.

Examples for XML Requests and Answers

Example Request(list inquire)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0103125</code>
    <view>
      <offset>0</offset>
      <limit>10</limit>
      <children>true</children>
      <from>2014-01-01 00:00:00</from>
      <to>2014-06-03 23:59:59</to>
    </view>
    <where>
      <key>domain</key>
      <operator>like</operator>
      <value>*</value>
    </where>
    <key>registry_when_date</key>
    <key>type</key>
    <key>owner</key>
    <key>reply_to</key>
    <key>ctid</key>
    <key>gaining_registrar</key>
    <key>disconnect</key>
    <key>spool_when</key>
    <key>changed</key>
    <key>created</key>
    <key>domain</key>
    <key>updater</key>
  </task>
</request>

```

Example Response (list inquire)

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>2</summary>
      <cancelation>
        <execdate>expire</execdate>
        <changed>2014-06-03 14:46:17</changed>
        <domain>example.com</domain>
        <type>DELETE</type>
        <spool_when>2015-06-03 02:45:51</spool_when>
        <registry_when_date>2015-06-03 14:45:51</registry_when_date>
        <log_id>32539</log_id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updater>
          <user>customer</user>
          <context></context>
        </updater>
        <created>2014-06-03 14:46:17</created>
      </cancelation>
      <cancelation>
        <execdate>expire</execdate>
        <changed>2014-06-03 14:46:27</changed>
        <domain>example2.com</domain>
        <type>DELETE</type>
        <spool_when>2015-06-03 02:45:53</spool_when>
        <registry_when_date>2015-06-03 14:45:53</registry_when_date>
        <log_id>32540</log_id>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updater>
          <user>customer</user>
          <context></context>
        </updater>
        <created>2014-06-03 14:46:27</created>
      </cancelation>
    </data>
    <status>
      <code>S0103125</code>
      <text>Domain cancelation request history has been inquired.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20140603-app1-dev-4635</stid>
</response>

```

6.11 Domain Restore (0101005)

With this task you can restore deleted domains.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0"encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0101005</code>
    <domain>
      <name>example.com</name>
    </domain>
    <reply_to>customer@example.com</reply_to>
    <ctid>ABCDEFGH</ctid>
  </task>
</request>
```

Example Response

```
<response version="1">
  <result>
    <data />
    <status>
      <code>N0101005</code>
      <type>notify</type>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
</response>
```

Example for a domain restore request with simultaneous renewal of the domain registration period (Renew)

Example Request

```
<?xml version="1.0"encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0101005</code>
    <domain>
      <name>example.com</name>
      <action>RESTORE_RENEW</action>
    </domain>
    <reply_to>customer@example.com</reply_to>
    <ctid>ABCDEFGH</ctid>
  </task>
</request>
```

Example Response

```
<response version="1">
  <result>
    <data></data>
    <status>
      <code>N0101005</code>
      <type>notify</type>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
</response>
```

6.12 Domain Restore List (0105005)

With this task you can list all canceled domains that can be restored.

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
action	Possible restore type. <ul style="list-style-type: none"> ▪ RESTORE_ARGP (= Auto Renew Grace Period) will be restored free of charge. ▪ RESTORE_RGP (= Redemption Grace Period) is associated with costs. 			yes		yes	yes
end_of_phase	End of the RGP restore phase.			yes			
name	Domainname			yes		yes	yes
owner	Object owner			yes		yes	yes
adminc	Domain Administrator				yes	yes	yes
authinfo	AuthInfo				yes	yes	yes

Value	Description	Info	Info <key>	List	List <key>	order	where
created	Creation date of the restore request				yes	yes	yes
expire	End of registration period				yes	yes	yes
nserver	Domain nameserver				yes	yes	yes
ownerc	Domaininhaber				yes	yes	yes
payable	Due date				yes	yes	yes
registry_status	Domain status at the registry				yes	yes	yes
techc	Technical administrator				yes	yes	yes
zonec	Zone administrator				yes	yes	yes

IMPORTANT NOTE

Please be aware that only domains marked with the action RESTORE_ARGP (= Auto Renew Grace Period) will be restored free of charge. A domain restore for domains marked with the action RESTORE_RGP (= Redemption Grace Period) are always associated with costs. Please see the current price list.

Examples for XML Requests and Answers**Example Request**

```

<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0105005</code>
    <where>
      <key>name</key>
      <operator>like</operator>
      <value>*</value>
    </where>
    <view>
      <offset>0</offset>
      <limit>3</limit>
      <children>0</children>
    </view>
    <key>owner</key>
    <key>created</key>
    <key>payable</key>
    <key>ownerc</key>
    <key>adminc</key>
    <key>techc</key>
    <key>zonec</key>
    <key>nserver</key>
    <key>authinfo</key>
    <key>registry_status</key>
    <key>expire</key>
    <ctid>ABCDEFG</ctid>
  </task>
</request>

```

Example Response

```

<response version="1">
  <result>
    <data>
      <summary>10</summary>
      <restore>
        <name>example.com</name>
        <expire>2011-12-23 15:35:54</expire>
        <payable>2012-02-15 15:44:41</payable>
        <ownerc>10458922</ownerc>
        <adminc>10458922</adminc>
        <techc>10458922</techc>
        <zonec>10458922</zonec>
        <registry_status>LOCK</registry_status>
        <nserver>
          <name>ns1.example.com</name>
        </nserver>
        <nserver>
          <name>ns2.example.com</name>
        </nserver>
        <action>RESTORE_ARGP</action>
        <authinfo>Y8HKJXgCupOZAWx</authinfo>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <created>2012-08-15 15:44:41</created>
      </restore>
      <restore>
        <name>example2.com</name>
        <expire>2011-12-23 15:35:54</expire>
        <payable>2012-02-15 15:44:41</payable>
        <ownerc>10458922</ownerc>
        <adminc>10458922</adminc>
        <techc>10458922</techc>
        <zonec>10458922</zonec>
        <registry_status>LOCK</registry_status>
        <nserver>
          <name>ns1.example.com</name>
        </nserver>
        <nserver>
          <name>ns2.examlple.com</name>
        </nserver>
        <action>RESTORE_ARGP</action>
        <authinfo>Y8HKJXgCupOZAWx</authinfo>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <created>2011-02-15 15:44:41</created>
      </restore>
      <restore>
        <name>example3.com</name>
        <expire>2011-12-23 15:35:54</expire>
        <payable>2012-02-15 15:44:41</payable>
        <ownerc>10458922</ownerc>
        <adminc>10458922</adminc>
        <techc>10458922</techc>
        <zonec>10458922</zonec>
        <registry_status>LOCK</registry_status>
        <nserver>
          <name>example.com</name>
        </nserver>
        <nserver>
          <name>ns2.example.com</name>
        </nserver>
        <action>RESTORE_ARGP</action>

```



```
<authinfo>Y8HKJXgCupOZAWx</authinfo>
  <owner>
    <user>customer</user>
    <context></context>
  </owner>
  <created>2011-02-15 15:44:41</created>
</restore>
</data>
<status>
  <code>S0105005</code>
  <type>success</type>
</status>
</result>
</response>
```

7 Domain Transfer

7.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	DomainTransferIn 0104 Domain Transfer Plus 0104 Domain Transfer Plus with Ownerchange 0104010 Domain Status 0102002 DomainTransferOut 010600 TransferOutInquire 0106001 AuthInfo Create 0113001 AuthInfo Delete 0113002 AuthInfo2 Create 0113003 IRTP Domain Info 0104008 IRTP Domain Restart 0114002
domain	1 - n		
domain / name	1	string	Name of the domain for which the respective task is carried out.
domain / ignore_whois	0 - 1		1 = starts an internal transfer even though the losing reseller did not update the status to "ACTIVE". NOTE You always need an AuthInfo to start any type of transfer.
domain / owncrc	0 - 1*	contact	Owner contact ID or the complete contact block
domain / adminc	0 - 1 *	contact	Administrative contact ID or the complete contact blocks.
domain / techc	0 - 1 *	contact	Technical contact ID or the complete contact block
domain / zonec	0 - 1 *	contact	Zone contact ID or the complete contact bloc
domain / nserver	1 - 8 *		
domain / nserver / name	1	domain	Host name of the nameserver
domain / nserver / ip	0 - 1 *	ip	Nameserver IP only for GLUE records; here it is mandatory NOTE GLUE records end with the domain name. In case you are using GLUE records you have to send the <ip> element containing the IPv4-address of the nameserver additionally.
domain / nserver / ip6	0 - 1 *	ip6	IPv6 of the nameserver, only for GLUE records;optional. NOTE The IPv6 nameserver can only be entered if an IPv4 value

Key	Occurs	Format	Description
			is also entered.
domain / nsentry	0 - 7 *	nsentry	Nsentry is only available for .DE. If nsentry is used, nserver is not allowed.
domain / nscheck	0 - 1	enum	An additional nameserver check is processed <ul style="list-style-type: none"> • soa = only soa values will be checked • ns = only nameservers will be checked • all = soa and ns will be checked
domain / zone / ip	0 - 1		IP address of the zone (A Record)
domain / zone / mx	0 - n		MX records. Enter the complete domain name.
domain / zone / ns_action	0 - 1	string	<p><u>Nameserver Mode</u></p> <ul style="list-style-type: none"> • complete: the zone will be generated on all nameservers • primary: the zone will only be generated on the primary nameserver • secondary: the zone will only be generated on the secondary nameserver. Zone data will be transferred to the primary nameserver via AXFR • hidden primary: the zone will be generated on the primary nameserver. The zone data will be transferred to the secondary nameserver via AXFR. The primary nameserver is not displayed as a nameserver for the zone.
domain / registry_status	0 - 1	enum	<p>Sets the domain status after the transfer</p> <p><u>The statuses</u></p> <ul style="list-style-type: none"> • LOCK = the domain is protected against an outgoing transfer. • HOLD = the domain is registered but not connected. (Example: resellers can set this status e.g. if a client did not pay.) • HOLD-LOCK= the domain is registered but not connected. It is also protected against an outgoing transfer. (Example: resellers can set this status e.g. if a client did not pay.) • ACTIVE = domain will be unlocked. <p>Default: LOCK</p>
domain / authinfo	0 - 1 *		AuthInfo for the domain
domain / use_privacy	0 - 1	enum	1= true You can also enter "true" as value. The key is optional and only available for some domains. You can use the task <i>user inquire</i> to inquire the respective tasks.
domain / use_trustee	0 - 1	enum	1= true You also can use the word "true" as valid value.

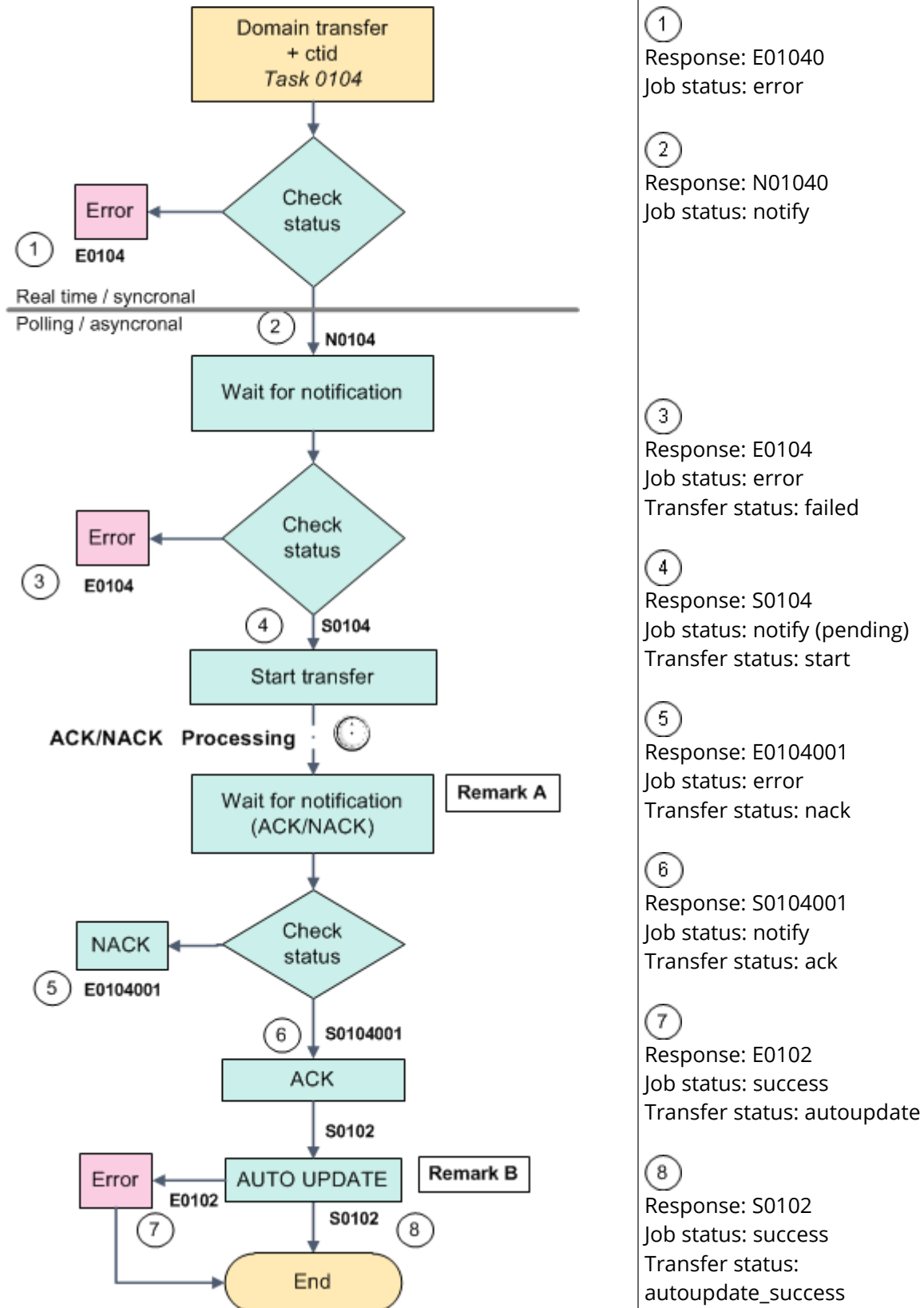
Key	Occurs	Format	Description
domain / irtp	0 - 1		IRTP data NOTE In the user profile, you can define the default entries for the IRTP data.
domain / irtp / foa1	1		FOA 1 data.
domain / irtp / foa1 / agreement_link	1	url	FOA1 agreement link
domain / irtp / foa1 / confirm_link	1	url	The confirmation link for the acknowledgement of the transfer.
domain / irtp / foa1 / from	1	string	Name of the sender, max. 50 characters. At the end „/ Domain Robot“ is automatically added.
domain / irtp / foa1 / reminder	1	enum	<ul style="list-style-type: none"> • yes • no Default: yes
domain / irtp / foa1 / supportmail	1	email	Support email address
domain / irtp / foa1 / to	1	enum	The recipient of the FOA1 email. <ul style="list-style-type: none"> • owner • admin • both (= OwnerC and AdminC) Default: both
domain / confirm_order	0 - 1	int	Some TLDs require a confirmation of the terms and conditions of registration. 1 = confirmation For other TLDs the key will be ignored.
default	0 - 1		Default-Block. The block from which the domain blocks in bulk transfers take their values for name and AuthInfo from. If you enter the name and AuthInfo tags into a domain block, the default values will be overwritten for that block.
execdate	0 - 1	execdate	Time of execution. Format: yyyy-MM-dd optional HH:mm:ss
domain / dnssec	0 - n		NOTE DNSSEC is still in a test phase at the moment. We therefore allow ourselves to make changes to our API at any time any without prior notice. DNSSEC can not be used with our nameservers at this time.
domain / dnssec / flags		enum	257 = KSK 256 = ZSK
domain / dnssec / protocol		int	Value: 3 (DNSSEC)
domain / dnssec / algorithm		enum	1= RSA/MD5 2= Diffie Hellman 3= DSA/SHA-1

Key	Occurs	Format	Description
			4= Elliptische Kurven 5= RSA/SHA-1 7= RSASHA1-NSEC3-SHA1 8= RSA/SHA-256 10= RSA/SHA-512 12= GOST R 34.10-2001
domain / dnssec / key			Key-signing-key ("KSK") that is to be transferred to the registry.
domain / dnssec / publickey		Base64	Base 64 coded public key
irtp	1		Data of the transferred domain
irtp / domain	1	string	Name of the transferred domain
transfer / domain	1	string	Name of the domain (Transfer Out)
transfer / type	1	string	Transfer request response ack= acknowledgement nack = no acknowledgement (Transfer Out)
transfer / nack_reason	0 -1	string	Required for type „nack“: <u>Available values and their meanings:</u> 1 = fraud. Evidence of fraud 2 = UDRP. UDRP action 3 = lawsuit. Court order by a court of competent jurisdiction 4 = identity. Reasonable dispute over the identity of the transfer contact 5 = nonpayment. No payment for previous registration period 6 = protest. Express written objection to the transfer from the transfer contact

*) Depends on the registry.

7.2 Domain Transfer In (0104)

With this task you can start an incoming domain transfer.



- ① Response: E0104
Job status: error
- ② Response: N0104
Job status: notify
- ③ Response: E0104
Job status: error
Transfer status: failed
- ④ Response: S0104
Job status: notify (pending)
Transfer status: start
- ⑤ Response: E0104001
Job status: error
Transfer status: nack
- ⑥ Response: S0104001
Job status: notify
Transfer status: ack
- ⑦ Response: E0102
Job status: success
Transfer status: autoupdate
- ⑧ Response: S0102
Job status: success
Transfer status: autoupdate_success

NOTE

The *OwnerChange* task can only be used for some TLDs. You can use the *DomainTransfer* task to start a domain transfer in combination with an *OwnerChange* task. Some registries call this combination of tasks a trade.

Examples for XML Requests and Answers**Example Request(single request)**

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code></code>
    <domain>
      <name>example.com</name>
      <authinfo>9Vz6U72V/RycSZW</authinfo>
      <ignore_whois>1</ignore_whois>
      <ownerc>9925345</ownerc>
      <adminc>9925345</adminc>
      <techc>9925345</techc>
      <zonec>9925345</zonec>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
      <dnssec>
        <flags>257</flags>
        <protocol>3</protocol>
        <algorithm>5</algorithm>
        <publickey>key...</publickey>
      </dnssec>
      <use_privacy>1</use_privacy>
      <confirm_order>1</confirm_order>
    </domain>
    <reply_to>customer@example.com</reply_to>
    <ctid></ctid>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0104</code>
      <text>Domain transfer has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

Example Request(bulk request)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0104</code>
    <default>
      <ignore_whois>1</ignore_whois>
      <ownerc>9925345</ownerc>
      <adminc>9925345</adminc>
      <techc>9925346</techc>
      <zonec>9925346</zonec>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
    </default>
    <domain>
      <name>example.com</name>
      <authinfo>9Vz6U72V/RycSZW</authinfo>
    </domain>
    <domain>
      <name>example.org</name>
      <authinfo>2V/RycSZW9Vz6U7</authinfo>
    </domain>
    <ctid>1234567</ctid>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0104</code>
      <text>Domain transfer has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

7.3 Domain Transfer Plus (0104)

With this task you can transfer a domain and create a new domain contact (handle) in a single task. You enter the data of a domain contact (handle) and the domain contact will then be generated automatically. If a contact with this data already exists, the existing contact will be used. In this case no new contact will be generated.

Within the “domain” element you can use all keywords which are allowed for the “normal” DomainTransfer request.

In the respective domain contact element (ownerc, adminc, techc, zonec) you can use all keys, which are valid for creating a domain contact. Solely the “nic_ref” key cannot be used.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0104</code>
    <domain>
      <name>example.com</name>
      <ctid>x11111-1</ctid>
      <ownerc>
        <type>PERSON</type>
        <fname>Peter</fname>
        <lname>Doe</lname>
        <title></title>
        <organization>Example Company</organization>
        <address>Example Street 3</address>
        <pcode>12345</pcode>
        <city>Example City</city>
        <state></state>
        <country>US</country>
        <phone>+1-12345-127335</phone>
        <fax>+1-12345-12345</fax>
        <email>Doe@example.com</email>
        <sip></sip>
        <protection>B</protection>
        <nic_ref>info</nic_ref>
        <nic_ref>org</nic_ref>
        <remarks></remarks>
        <extension>
          <gender></gender>
          <idnumber></idnumber>
          <vatnumber></vatnumber>
          <trademark_name></trademark_name>
          <trademark_number></trademark_number>
          <trademark_country></trademark_country>
          <trademark_regdate></trademark_regdate>
          <trademark_appdate></trademark_appdate>
          <companynumber></companynumber>
          <birthday></birthday>
          <birthplace></birthplace>
          <birthcountry></birthcountry>
        </extension>
      </ownerc>
      <adminc>
        <type>PERSON</type>
        <fname>Hans</fname>
        <lname>Doe</lname>
        <title></title>
        <organization>Example Company</organization>
        <address>Example Street 3</address>
        <pcode>12745</pcode>
        <city>Example City</city>
        <state></state>
        <country>US</country>
        <phone>+1-12345-166666</phone>
        <fax>+1-12345-127745</fax>
        <email>Doe@example.com</email>
        <sip></sip>
        <protection>B</protection>
      </adminc>
    </domain>
  </task>
</request>
```

```

<nic_ref>info</nic_ref>
<nic_ref>org</nic_ref>
<remarks></remarks>
<extension>
  <gender></gender>
  <idnumber></idnumber>
  <vatnumber></vatnumber>
  <trademark_name></trademark_name>
  <trademark_number></trademark_number>
  <trademark_country></trademark_country>
  <trademark_regdate></trademark_regdate>
  <trademark_appdate></trademark_appdate>
  <companynumber></companynumber>
  <birthday></birthday>
  <birthplace></birthplace>
  <birthcountry></birthcountry>
</extension>
</adminc>
<techc>
  <type>PERSON</type>
  <fname>Hans</fname>
  <lname>Doe</lname>
  <title></title>
  <organization>Example Company</organization>
  <address>Example Street 3</address>
  <pcode>12345</pcode>
  <city>Example City</city>
  <state></state>
  <country>US</country>
  <phone>+1 -12345-166666</phone>
  <fax>+1-12345-127745</fax>
  <email>Doe@example.com</email>
  <sip></sip>
  <protection>B</protection>
  <nic_ref>info</nic_ref>
  <nic_ref>org</nic_ref>
  <remarks></remarks>
  <extension>
    <gender></gender>
    <idnumber></idnumber>
    <vatnumber></vatnumber>
    <trademark_name></trademark_name>
    <trademark_number></trademark_number>
    <trademark_country></trademark_country>
    <trademark_regdate></trademark_regdate>
    <trademark_appdate></trademark_appdate>
    <companynumber></companynumber>
    <birthday></birthday>
    <birthplace></birthplace>
    <birthcountry></birthcountry>
  </extension>
</techc>
<zonec>
  <type>PERSON</type>
  <fname>Hans</fname>
  <lname>Doe</lname>
  <title></title>
  <organization>Example Company</organization>
  <address>Example Street 3</address>
  <pcode>12345</pcode>
  <city>MusterstadtExample City</city>
  <state></state>
  <country>US</country>
  <phone>+1-12355-166666</phone>
  <fax>+1-12345-133345</fax>
  <email>Doe@example.com</email>
  <sip></sip>
  <protection>B</protection>

```

```

<nic_ref>info</nic_ref>
<nic_ref>org</nic_ref>
<remarks></remarks>
<extension>
  <gender></gender>
  <idnumber></idnumber>
  <vatnumber></vatnumber>
  <trademark_name></trademark_name>
  <trademark_number></trademark_number>
  <trademark_country></trademark_country>
  <trademark_regdate></trademark_regdate>
  <trademark_appdate></trademark_appdate>
  <companynumber></companynumber>
  <birthday></birthday>
  <birthplace></birthplace>
  <birthcountry></birthcountry>
</extension>
</zonec>      <nserver>
  <name>ns1.example.com</name>
</nserver>
<nserver>
  <name>ns2.example.com</name>
</nserver>
<!-- Mandatory for some TLDs.-->
<confirm_order>1</confirm_order>
<!-- Optional for some TLDsFür einige TLDs optional -->
<use_trustee>0</use_trustee>
</domain>
<reply_to>
  <email>MusterDoe@example.com</reply_to>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0104</code>
      <object>
        <type>owner-c</type>
        <value>9926617</value>
      </object>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
      <text>Domain transfer has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>

```

7.4 Domain Transfer Plus with Ownerchange (0104010)

With this task you can transfer a domain and change the OwnerC in a single task. You enter the complete data of the domain contact (handle) of the new OwnerC in the task. The domain contact will then be generated automatically. If a contact with this data already exists, the existing contact will be used. In this case no new contact will be generated.

Within the “domain” element you can use all keywords which are allowed for the “normal” DomainTransfer request.

In the respective domain contact element (ownerc) you can use all keys, which are valid for creating a domain contact. Solely the “nic_ref” key cannot be used.

NOTE

The *OwnerChange* task can only be used for some TLDs. You can use the *DomainTransfer* task to start a domain transfer in combination with an *OwnerChange* task. Some registries call this combination of tasks a trade.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0104010</code>
    <domain>
      <name>example.com</name>
      <ctid>x11111-1</ctid>
      <ownerc>
        <type>PERSON</type>
        <fname>Peter</fname>
        <lname>Doe</lname>
        <title></title>
        <organization>Example Company</organization>
        <address>Example Street 3</address>
        <pcode>12345</pcode>
        <city>Example City</city>
        <state></state>
        <country>US</country>
        <phone>+1-12345-127335</phone>
        <fax>+1-12345-12345</fax>
        <email>Doe@example.com</email>
        <sip></sip>
        <protection>B</protection>
        <nic_ref>info</nic_ref>
        <nic_ref>org</nic_ref>
        <remarks></remarks>
        <extension>
          <gender></gender>
          <idnumber></idnumber>
          <vatnumber></vatnumber>
          <trademark_name></trademark_name>
          <trademark_number></trademark_number>
          <trademark_country></trademark_country>
          <trademark_regdate></trademark_regdate>
          <trademark_appdate></trademark_appdate>
          <companynumber></companynumber>
        </extension>
      </ownerc>
    </domain>
  </task>
</request>
```

```

        <birthday></birthday>
        <birthplace></birthplace>
        <birthcountry></birthcountry>
    </extension>
</ownerc>
<adminc>
    <type>PERSON</type>
    <fname>Hans</fname>
    <lname>Doe</lname>
    <title></title>
    <organization>Example Company</organization>
    <address>Example Street 3</address>
    <pcode>12745</pcode>
    <city>Example City</city>
    <state></state>
    <country>US</country>
    <phone>+1-12345-166666</phone>
    <fax>+1-12345-127745</fax>
    <email>Doe@example.com</email>
    <sip></sip>
    <protection>B</protection>
    <nic_ref>info</nic_ref>
    <nic_ref>org</nic_ref>
    <remarks></remarks>
    <extension>
        <gender></gender>
        <idnumber></idnumber>
        <vatnumber></vatnumber>
        <trademark_name></trademark_name>
        <trademark_number></trademark_number>
        <trademark_country></trademark_country>
        <trademark_regdate></trademark_regdate>
        <trademark_appdate></trademark_appdate>
        <companynumber></companynumber>
        <birthday></birthday>
        <birthplace></birthplace>
        <birthcountry></birthcountry>
    </extension>
</adminc>
<techc>
    <type>PERSON</type>
    <fname>Hans</fname>
    <lname>Doe</lname>
    <title></title>
    <organization>Example Company</organization>
    <address>Example Street 3</address>
    <pcode>12345</pcode>
    <city>Example City</city>
    <state></state>
    <country>US</country>
    <phone>+1 -12345-166666</phone>
    <fax>+1-12345-127745</fax>
    <email>Doe@example.com</email>
    <sip></sip>
    <protection>B</protection>
    <nic_ref>info</nic_ref>
    <nic_ref>org</nic_ref>
    <remarks></remarks>
    <extension>
        <gender></gender>
        <idnumber></idnumber>
        <vatnumber></vatnumber>
        <trademark_name></trademark_name>
        <trademark_number></trademark_number>
        <trademark_country></trademark_country>
        <trademark_regdate></trademark_regdate>
        <trademark_appdate></trademark_appdate>
        <companynumber></companynumber>
        <birthday></birthday>
        <birthplace></birthplace>
        <birthcountry></birthcountry>
    </extension>
</techc>

```

```

    <zonec>
      <type>PERSON</type>
      <fname>Hans</fname>
      <lname>Doe</lname>
      <title></title>
      <organization>Example Company</organization>
      <address>Example Street 3</address>
      <pcode>12345</pcode>
      <city>MusterstadtExample City</city>
      <state></state>
      <country>US</country>
      <phone>+1-12355-166666</phone>
      <fax>+1-12345-133345</fax>
      <email>Doe@example.com</email>
      <sip></sip>
      <protection>B</protection>
      <nic_ref>info</nic_ref>
      <nic_ref>org</nic_ref>
      <remarks></remarks>
      <extension>
        <gender></gender>
        <idnumber></idnumber>
        <vatnumber></vatnumber>
        <trademark_name></trademark_name>
        <trademark_number></trademark_number>
        <trademark_country></trademark_country>
        <trademark_regdate></trademark_regdate>
        <trademark_appdate></trademark_appdate>
        <companynumber></companynumber>
        <birthday></birthday>
        <birthplace></birthplace>
        <birthcountry></birthcountry>
      </extension>
    </zonec>
  </nserver>
  <nserver>
    <name>ns1.example.com</name>
  </nserver>
  <nserver>
    <name>ns2.example.com</name>
  </nserver>
  <!-- Mandatory for some TLDs.-->
  <confirm_order>1</confirm_order>
  <!-- Optional for some TLDs. -->
  <use_trustee>0</use_trustee>
</domain>
  <reply_to>
    <email>MusterDoe@example.com</reply_to>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0104</code>
      <object>
        <type>owner-c</type>
        <value>9926617</value>
      </object>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
      <text>Domain transfer has been started successfully.</text>
      <type>notify</type>
    </status>
  </result>
</response>

```

7.5 Domain Transfer-Out (0106002)

With this task you can answer a current outgoing transfer (ACK/ NACK).

Examples for XML Requests and Answers

Example Request(ACK)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0106002</code>
    <transfer>
      <domain>example.com</domain>
      <type>ack</type>
    </transfer>
  </task>
</request>
```

Example Response(ACK) N0106

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0106</code>
      <text>The response to the domain transfer request has been started.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

Example Request(NACK)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0106002</code>
    <transfer>
      <domain>example.com</domain>
      <type>nack</type>
      <nack_reason>7</nack_reason>
    </transfer>
  </task>
</request>
```

Example Response(NACK) N0106

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0106</code>
```

```

    <text>The response to the domain transfer request has been started.</text>
    <type>notify</type>
  </status>
</result>
</response>

```

7.6 Transfer-Out Inquire Tasks (0106001)

With this task you can inquire the data of the current outgoing transfers, to be answered with ACK or NACK. With the single inquire you receive the data of a single transfer-out request. With the list inquire you receive a list of transfers to be answered with ACK.

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
created	Creation date of the transfer	yes		yes		yes	yes
domain	The domain that is to be transferred	yes		yes		yes	yes
end	Transfer end	yes			yes	yes	yes
gaining_registrar	Future registrar	yes			yes	yes	yes
loosing_registrar	Current registrar	yes			yes	yes	yes
owner	Object owner	yes		yes		yes	yes
reminder	Sending date of reminder (FOA2) has been sent on.	yes			yes	yes	yes
start	Transfer start	yes			yes	yes	yes
transaction	Client transaction ID which was set in the transfer answer.	yes			yes	yes	yes
type	Transfer request response ack= acknowledgement nack = no acknowledgement	yes					
nack_reason	Reason for rejection (NACK)	yes					

Value	Description	Info	Info <key>	List	List <key>	order	where
auto_nack	Date on which the Registry triggered the AutoNACK.	yes					
updated	Last updated	yes		yes		yes	yes

7.7 TransferOut Info (0106001)

With this task you can inquire the data of one currently outgoing transfer, to be answered with ACK or NACK.

Examples for XML Requests and Answers

Example Request(single inquire)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0106001</code>
    <transfer>
      <domain>
        <name>example.com</name>
      </domain>
    </transfer>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <transfer>
        <transaction>1213629750</transaction>
        <type>ack</type>
        <nack_reason>0</nack_reason>
        <domain>example.com</domain>
        <gaining_registrar>regiatarA</gaining_registrar>
        <loosing_registrar>RegistrarB</loosing_registrar>
        <start>2009-06-16 14:52:23</start>
        <reminder>2009-06-19 14:52:23</reminder>
        <autonack>2009-06-20 18:52:23</autonack>
        <end>2009-06-21 14:52:23</end>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updater>
          <user>example.com</user>
          <context></context>
        </updater>
        <id>3794</id>
        <created>2009-06-16 17:22:07</created>
        <updated>2012-06-12 13:24:35</updated>
      </transfer>
```

```

    </data>
    <status>
      <code>S0106001</code>
      <text>Transfer data have been inquired successfully.</text>
      <type>success</type>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
</response>

```

7.8 TransferOut List (0106001)

With this task you can inquire the data of several currently outgoing transfers, to be answered with ACK or NACK.

Examples for XML Requests and Answers

Example Request(list inquire)

```

<?xml version="1.0" encoding="UTF-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0106001</code>
    <view>
      <children>true</children>
    </view>
    <key>start</key>
    <key>end</key>
    <key>gaining_registrar</key>
    <key>loosing_registrar</key>
    <key>reminder</key>
    <key>transaction</key>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <transfer>
        <transaction>1213629750</transaction>
        <domain>example.com</domain>
        <gaining_registrar>registrarA</gaining_registrar>
        <loosing_registrar>registrarB</loosing_registrar>
        <start>2009-06-16 14:52:23</start>
        <reminder>2009-06-19 14:52:23</reminder>
        <end>2009-06-21 14:52:23</end>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <created>2009-06-16 17:22:07</created>
        <updated>2012-06-12 13:24:35</updated>
      </transfer>
    </data>
  </result>
</response>

```

```

</transfer>
<transfer>
  <transaction>1213629751</transaction>
  <domain>example1.com</domain>
  <gaining_registrar>registrarA</gaining_registrar>
  <loosing_registrar>registrarB</loosing_registrar>
  <start>2009-06-16 14:52:23</start>
  <reminder>2009-06-19 14:52:23</reminder>
  <end>2009-06-21 14:52:23</end>
  <owner>
    <user>admin_8</user>
    <context>1</context>
  </owner>
  <created>2009-06-16 17:22:07</created>
  <updated>2012-06-12 13:24:35</updated>
</transfer>
<transfer>
  <transaction>1213629752</transaction>
  <domain>example2.com</domain>
  <gaining_registrar>registrarA</gaining_registrar>
  <loosing_registrar>registrarB</loosing_registrar>
  <start>2009-06-16 14:52:23</start>
  <reminder>2009-06-19 14:52:23</reminder>
  <end>2009-06-21 14:52:23</end>
  <owner>
    <user>customer</user>
    <context></context>
  </owner>
  <created>2009-06-16 17:22:07</created>
  <updated>2015-06-10 08:35:08</updated>
</transfer>
</data>
<status>
  <code>S0106001</code>
  <text>Transfer data have been inquired successfully.</text>
  <type>success</type>
</status>
</result>
</response>

```

7.9 Domain Status (0102002)

With this task you can set the status of your domain. (LOCK, HOLD, HOLD-LOCK, ACTIVE). Some registries allow you to lock your domains to protect them against transfer. Only gTLDs have a domain status. Please note, that some registries do not support all statuses

The statuses

- LOCK = the domain is protected against an outgoing transfer.
- HOLD = the domain is registered but not connected. (Example: resellers can set this status e.g. if a client did not pay.)
- HOLD-LOCK= the domain is registered but not connected. It is also protected against an outgoing transfer. (Example: resellers can set this status e.g. if a client did not pay.)
- ACTIVE = domain will be unlocked.

Default: LOCK

Examples for XML Requests and Answers

Example Request(single request)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0102002</code>
    <domain>
      <name>example.com</name>
      <registry_status>ACTIVE</registry_status>
    </domain>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data></data>
    <status>
      <code>N0102003</code>
      <text>Domain status update has been started.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

Example Request(bulk request)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0102002</code>
    <default>
      <registry_status>lock</registry_status>
    </default>
    <domain>
      <name>example.com</name>
    </domain>
    <domain>
      <name>example.net</name>
    </domain>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data></data>
    <status>
      <code>N0102003</code>
```

```

        <text>Domain status update has been started.</text>
        <type>notify</type>
    </status>
</status>
    <code>N0102003</code>
    <text>Domain status update has been started.</text>
    <type>notify</type>
</status>
</result>
</response>

```

7.10 AuthInfo1Create (Apply for AuthInfo) (0113001)

With this task you can apply for an AuthInfo1.

NOTE

If a cancelation of the type preack exists you cannot generate an AuthInfo1 using the system. In this case the system generates the AuthInfo automatically. For the target dates „now“ and „date“ the AuthInfo1 is valid 30 days from the cancelation date (execdate) onwards. Selecting the target date „expire“, i.e. cancelation on the expiration date, the AuthInfo1 validity starts 30 days before the cancelation date (execdate).

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <owner>
    <user>customer2</user>
    <context></context>
  </owner>
  <task>
    <code>0113001</code>
    <domain>
      <name>example.com</name>
    </domain>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data/>
    <status>
      <code>S0113001</code>
      <text>AuthInfo1 was created successfully.</text>
      <type>success</type>
      <object>
        <type>authinfo</type>
        <value>kiSV/CbJi9/KaCxp</value>
      </object>
    </status>
  </result>
</response>

```

Example Response, if an AuthInfo already exists

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data/>
    <msg>
      <text>For this domain an AuthInfo already exists.</text>
      <code>EF011305</code>
      <type>error</type>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
    </msg>
    <status>
      <code>E0113001</code>
      <text>AuthInfo1 could not be created.</text>
      <type>error</type>
    </status>
  </result>
</response>
```

Beispielantwort, nachdem jetzt der Auftrag ein zweites Mal gesendet wurde

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data/>
    <msg>
      <text>For this domain an AuthInfo already exists.</text>
      <code>EF011305</code>
      <type>error</type>
      <object>
        <type>domain</type>
        <value>example.com</value>
      </object>
    </msg>
    <status>
      <code>E0113001</code>
      <text>AuthInfo1 could not be created.</text>
      <type>error</type>
      <object>
        <type>authinfo</type>
        <value>12345-abcde</value>
      </object>
    </status>
  </result>
</response>
```

7.10.1 DENIC Domain Transfer Process with AuthInfo

In the DENIC transfer procedure a password, the AuthInfo, is used. It is generated exclusively for the transfer of a specific domain.

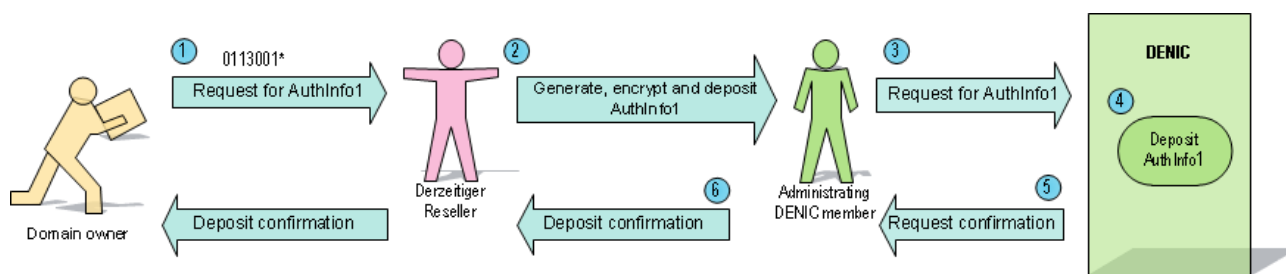
The AuthInfo is transferred and stored encoded at the DENIC database. It is assigned to one domain and valid only for 30 days. After this period the domain owner has to apply for a new AuthInfo.

The domain transfer is carried out in two steps:

1. Applying for AuthInfo, generating it and storing it at the DENIC database. There are two procedures available: AuthInfo is applied for at the current reseller (AuthInfo1) or, if that's not possible, it can be requested from any DENIC member (AuthInfo2).
2. Starting and carrying out the domain transfer via the new reseller.

The applied AuthInfo can also be deleted with a corresponding request.

Details on Process: Applying for AuthInfo at the Current Reseller, Generate and Storing it at DENIC (AuthInfo1)



- | | |
|---|---|
| ① | The domain owner requests AuthInfo from the current reseller. *Taskcode, if we are the reseller. |
| ② | The reseller tasks the current DENIC member with the generation and encryption of AuthInfo1 and its deposition at the DENIC. This is a main safety feature of the new AuthInfo procedure. |
| ③ | The current administrating DENIC member passes on the encrypted AuthInfo1 to the DENIC and requests for its deposition. |
| ④ | DENIC stores the encrypted AuthInfo1 in its database. The DENIC does not know the plain text of AuthInfo – a further important security feature of the new procedure. |
| ⑤ | DENIC sends a confirmation of the deposition request to the administrating DENIC member. |
| ⑥ | The administrating DENIC member sends the confirmation to the current reseller, who redirects it to the domain owner. |

7.11 AuthInfo Delete (0113002)

With this task you can delete an AuthInfo1 from a domain. The deletion of AuthInfo1 can be requested until the transfer starts. After that a deletion is no longer possible.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0113002</code>
    <domain>
      <name>example.de</name>
    </domain>
  </task>
</request>

```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0113002</code>
      <text>AuthInfo1 was deleted successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

7.12 AuthInfo2 Create (Apply for AuthInfo2) (0113003)

NOTE:

This process can also be carried out for the TLDs .DE, .EU, .NO and .BE. For .BE domains only the AuthInfo2 task can be used to get an AuthInfo for the domain transfer.

Examples for XML Requests and Answers

Example Request

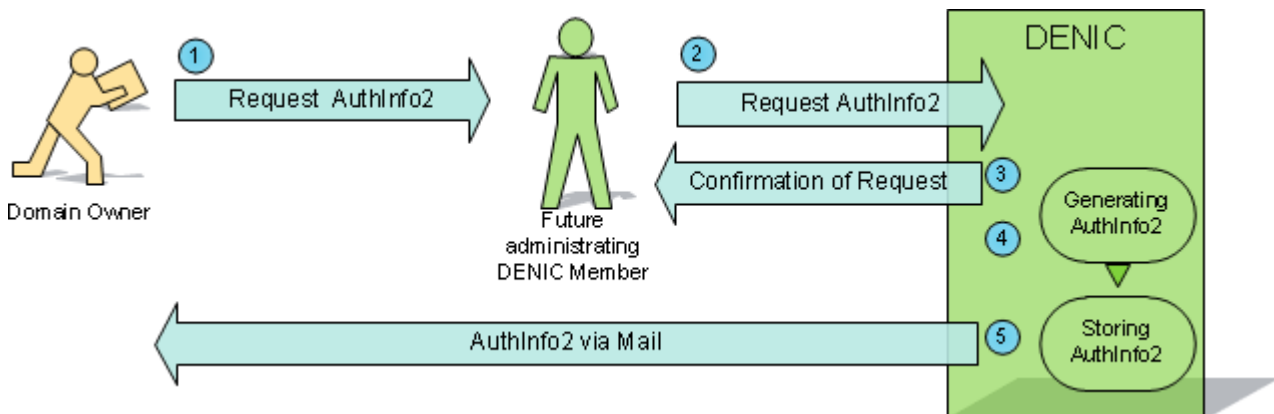
```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0113003</code>
    <domain>
      <name>example.de</name>
    </domain>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0113003</code>
      <text>AuthInfo2 was created successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```


7.12.1 Workflow for .DE: Requesting AuthInfo from the Future Administrating DENIC Member (AuthInfo2)

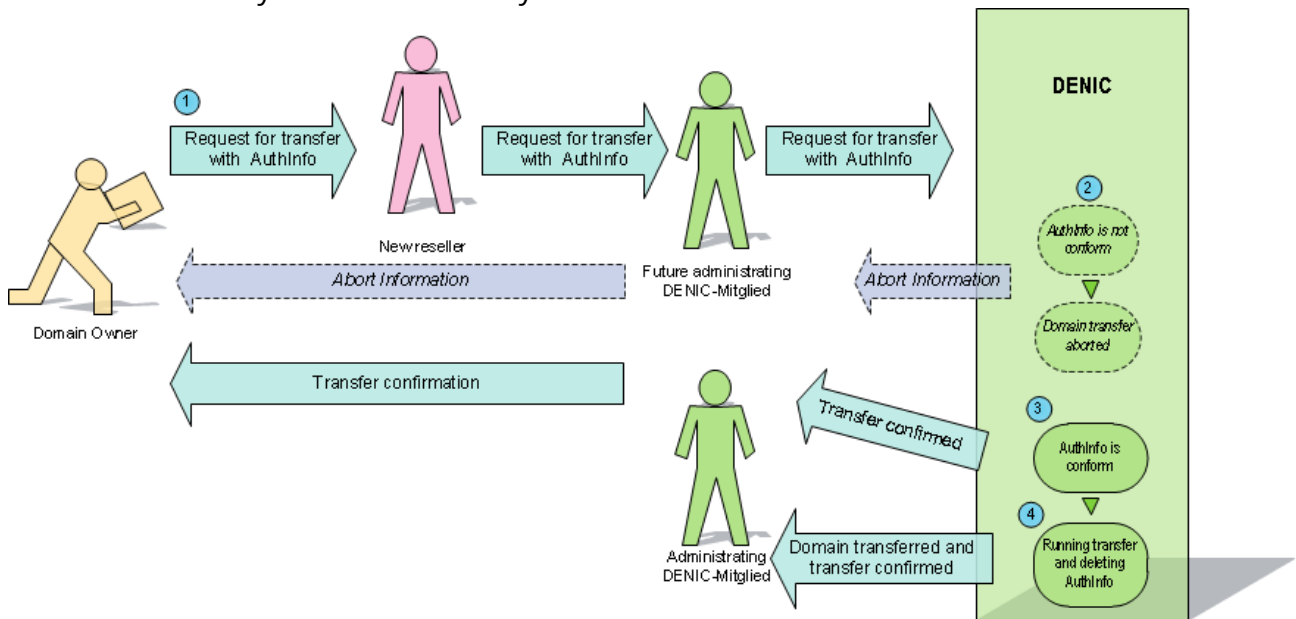
The domain owner can apply for AuthInfo2 directly at the DENIC member, who will administrate the domain in future or at any other DENIC member. That's convenient in case it might not be possible to apply for AuthInfo1 at the current reseller, e.g. because the reseller does no longer exist, is unreachable or does not respond.



1	The domain owner applies for AuthInfo2 directly at the DENIC member, who will administrate the domain in future.
2	DENIC member sends a request to the DENIC to generate and store AuthInfo2 in the DENIC database.
3	DENIC sends a confirmation of the request to the administrating member.
4	DENIC generates and stores AuthInfo2 in its database.
5	DENIC sends the AuthInfo2 via mail directly to the domain owner.
	The AuthInfo2 procedure is always associated with costs.

7.12.2 Example Process .DE: Starting the Domain Transfer

With the AuthInfo you can successfully initiate the transfer of the domain.



- | | |
|---|---|
| ① | The domain owner passes the AuthInfo on to the new reseller and orders the domain transfer.
Before the Transfer starts DENIC checks if an AuthInfo is stored for the domain . If no AuthInfo is stored, the transfer will not start.
If an AuthInfo is stored, the transfer starts. DENIC will compare the incoming and the stored AuthInfo for its authenticity. |
| ② | If the authenticity of AuthInfo is not proven the domain transfer will be aborted. The future administrative member will be informed about the abort. He passes on this information to the new reseller. |
| ③ | If the authenticity of AuthInfo is proven the domain transfer will be processed and AuthInfo will be deleted afterwards. |
| ④ | The involved DENIC members will be informed about the successful transfer. |

7.13 IRTP Domain Info (0104008)

With this task you can inquire the data of a currently active IRTP transfer.

This table contains all values that are returned for the List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- Value: name of the returned value
- Description: what does this value mean.
- Info <key>: The value can be inquired additionally in an Info task by using the <key> key.

Value	Description	Info	Info <key>
admin_email	Email of the administrative contact.	yes	
agreement_link	Liok to the IRTP agreement	yes	
cancel	Date, when the transfer was refused (nack).	yes	
created	Creation date of the transfer request	yes	
confirm_link	Confirmation link for the request	yes	
from		yes	
domain	Domain name	yes	
irtp_id	ID of the current IRTP job	yes	
loosing_registar	Current registrar	yes	
mail_log	Information about sent emails		
nic_member_label		yes	
owner	Object owner	yes	
owner_name	Object owner name	yes	
reminder	How often a reminder email was sent (FOA2)	yes	
responding_server		yes	
result	Transfer result	yes	
spool_id	ID for the request in the spool for internal processing.	yes	
status	Aktueller transfer status (CANCEL; TRANSFER)	yes	
status_history	Transfer status history	yes	
supportmail	Support email address	yes	
to	Recipients of FOA email reminder (OwnerC or AdminC and OwnerC (=both))	yes	
updated	Last updated	yes	

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <context></context>
    <password>password</password>
  </auth>
  <task>
    <code>0104008</code>
    <irtp>
      <irtp_id>612b94ad7918a2199e087365ae2d6e12</irtp_id>
    </irtp>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <irtp>
        <spool_id>195611805</spool_id>
        <irtp_id>6f7alc9deb94ec10b34a3d2ef8de835b</irtp_id>
        <domain>example.com</domain>
        <status>PENDING_PROGRESS</status>
        <owner_name>customer</owner_name>
        <email/>
        <admin_email/>
        <responding_server/>
        <loosing_registrar></loosing_registrar>
        <cancel>2015-08-25 00:00:00</cancel>
        <result>
          <code>IRTP0000</code>
          <msg>OK</msg>
        </result>
        <status_history>
          <status>
            <id>886131</id>
            <type>foal</type>
            <smtp_code>250</smtp_code>
            <msg>Enter mail, end with "."</msg>
            <status>success</status>
            <date>2015-08-11 14:11:09</date>
          </status>
        </status_history>
        <owner>
          <user>customer</user>
          <context>4</context>
        </owner>
        <created>2015-08-11 00:00:00</created>
        <updated>2015-08-11 14:11:09</updated>
        <nic_member_label>XXX-TEST</nic_member_label>
        <agreement_link>http://example.com/agreement_link</agreement_link>
        <confirm_link></confirm_link>
        <from>Domain Transfer / Domain Robot / Domain Robot</from>
        <reminder>0</reminder>
        <supportmail>support@example.com</supportmail>
        <to>both</to>
      </irtp>
    </data>
  </result>
</response>
```

```

    <code>S0104008</code>
    <text>Die IRTP-Daten wurden erfolgreich ermittelt.</text>
    <type>success</type>
  </object>
    <type>irtp</type>
    <value>example.com</value>
  </object>
</status>
</result>
<stid>20150811-app1-test-5822</stid>
</response>

```

7.14 Irtp DomainRestart (114002)

This task allows you to resend the FOA1 email to the registrant.

Examples for XML Requests and Answers

Example Request

```

<request>
  <auth>
    <user>Customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0114002</code>
    <spool_id>123456789</spool_id>
    <!-- You can use the domain name instead of the spool_id too.
  <domain>
    <name>example.com</name>
  </domain>
  -->
    <!-- For this task a bulk request is existing too. In the bulk request you can
    use the spool_id and the domain name. You also can merge them.-->
  </task>
</request>

```

Example Response

```

<response>
  <result>
    <status>
      <code>S0114002</code>
      <text>IRTP update successful.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

8 Domain Backorder

Since August 2011, a backorder process is available that allows to register .CO domains that are entering the "Pending Delete" phase. The .CO registry launches a 15-day "Redemption Period" in case of deletion of a domain. Then, the status of the domain changes to a 5-day "pending delete" phase.

The .CO domains that are pending deletion ("Redemption Period") and .CO domains disposable for backordering are displayed 15 days prior to their deletion in the overview of the .CO backorders.

Domains that are not restored by the current registrar, may be ordered after the redemption period from any authorized registrar. Following a first-come-first-served basis, the first registrar who submits the backorder will receive the domain.

If the registration was successful, the domain will be assigned to you after the 5-day "pending delete" phase.

If the domain was not supposed to be deleted but to be restored, you will be notified in case you have provided a backorder for the domain.

In the user profile (task 1301014 user profile update) you can store the following backorder keys:

- backorder_owncrc
- backorder_adminc
- backorder_techc
- backorder_zonec
- backorder_nserver1
- backorder_nserver2
- backorder_nserver3
- backorder_nserver4
- backorder_nserver5
- backorder_nserver6
- backorder_nserver7

8.1 Backorder Create (0141)

With this task you can create a backorder request.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
```

```

<task>
  <code>0141</code>
  <domain_backorder>
    <name>example.com.co</name>
  </domain_backorder>
  <reply_to></reply_to>
  <ctid></ctid>
</task>
</request>

```

Example Response

```

<?xml version="1.0"?>
<response version="1">
  <result>
    <data/>
    <status>
      <code>S0141</code>
      <type>success</type>
      <object>
        <type>domain_backorder</type>
        <value>example.com.co</value>
      </object>
    </status>
  </result>
</response>

```

8.2 Backorder Delete (0143)

With this task you can cancel a backorder request. Please be aware that you will be billed for this service.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0143</code>
    <domain_backorder>
      <name>example.com.co</name>
    </domain_backorder>
  </task>
</request>

```

Example Response

```

<?xml version="1.0"?>
<response version="1">
  <result>
    <data/>
    <status>
      <code>S0143</code>
      <type>success</type>
      <object>
        <type>domain_backorder</type>

```

```

        <value>example.com.co</value>
      </object>
    </status>
  </result>
</response>

```

8.3 Domain Backorder User List

With this task you can inquire the backorder domains of a user and his subusers.

This table contains all values that are returned for the List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- Value: name of the returned value
- Description: what does this value mean.
- List: Wert wird bei List-Task standardmäßig zurückgegeben
- List <key>: The value can be inquired additionally in an List task by using the <key> key.
- Order: Sorting
- Where: Filtering

Value	Description	List	List <key>	order	where
adminc	Domain administrator		yes	yes	yes
available	Availability of the domain		yes	yes	yes
created	Creation date of the request	yes		yes	yes
name	Domain name			yes	yes
nscopy	NSentry, for .de domains only		yes	yes	yes
nserver	Nameserver		yes	yes	yes
ownerc	Domain owner		yes	yes	yes
period	Registration period of the desired domains		yes	yes	yes
scheduled_drop	Scheduled Drop		yes	yes	yes
techc	Technical contact		yes	yes	yes
updated	Date of the last update	yes		yes	yes
zonec	Zone administrator		yes	yes	yes

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0145</code>
    <view>
      <limit>10</limit>
      <children>1</children>
    </view>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>2</summary>
      <domain_backorder>
        <name>backorder-201109-0738.co</name>
        <available>2011-09-20 19:00:00</available>
        <scheduled_drop>2011-09-30 19:00:00</scheduled_drop>
        <created>2011-09-18 18:22:05</created>
        <updated>2011-09-30 11:14:45</updated>
      </domain_backorder>
      <domain_backorder>
        <name>backorder-201109-0739.co</name>
        <available>2011-09-20 19:00:00</available>
        <scheduled_drop>2011-09-30 19:00:00</scheduled_drop>
        <created>2011-09-18 19:22:04</created>
        <updated>2011-09-30 11:14:45</updated>
      </domain_backorder>
    </data>
    <status>
      <code>S0145</code>
      <text>The list of all own backorder entries was loaded.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20120703-app1-test-233</stid>
</response>
```

8.4 Domain Backorder List (0146)

With this task you can inquire the available backorder domains.

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- Value: name of the returned value
- Description: what does this value mean.
- List: Wert wird bei List-Task standardmäßig zurückgegeben
- List <key>: The value can be inquired additionally in an List task by using the <key> key.
- Order: Sorting
- Where: Filtering

Value	Description	List	List <key>	order	where
adminc	Domain administrator		yes	yes	yes
available	Availability of the domain		yes	yes	yes
created	Creation date of the request	yes		yes	yes
name	Domain name		yes	yes	yes
nsentry	NSentry, for .de domains only		yes	yes	yes
nserver	Nameserver		yes	yes	yes
ownerc	Domain owner		yes	yes	yes
period			yes	yes	yes
scheduled_drop	Scheduled Drop		yes	yes	yes
techc	Technical contact		yes	yes	yes
updated	Date of the last update	yes		yes	yes
zonec	Zone administrator		yes	yes	yes

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0146</code>
```

```

<view>
  <limit>10</limit>
</view>
<where>
  <and>
    <or>
      <key>domain</key>
      <operator>like</operator>
      <value>*012*.*</value>
    </or>
    <or>
      <key>domain</key>
      <operator>like</operator>
      <value>xn--*012*</value>
    </or>
  </and>
</where>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>0</summary>
    </data>
    <status>
      <code>S0146</code>
      <text>The list of all free backorder entries has been loaded.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

Example Request

```

<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0146</code>
    <view>
      <offset>0</offset>
      <limit>3</limit>
      <children>0</children>
    </view>
    <order>
      <key>name</key>
      <key>ownerc</key>
      <key>adminc</key>
      <key>techc</key>
      <key>zonec</key>
      <key>nserver</key>
      <key>period</key>
      <key>available</key>
      <key>scheduled_drop</key>
      <ctid/>
      <reply_to></reply_to>
    </order>
  </task>
</request>

```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response version="1">
  <result>
    <data>
      <summary>0</summary>
    </data>
    <status>
      <code>S0146</code>
      <text>The list of all free backorder entries was loaded.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

9 Zone

9.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	ZoneCreate 0201 Zone Update 0202 Zone Import 0204 Zone Delete 0203 Zone Inquire 0205
zone	1 - n		
zone / name	1	string	Zone name
zone /ns_action	0 - 1	enum	Additional nameserver check is proceeded. <ul style="list-style-type: none"> complete: the zone will be generated on all nameservers primary: the zone will be generated only on the primary nameserver secondary: the zone will be generated only on the secondary nameserver. Zone data will be transferred to primary via AXFR. hidden primary: the zone will be generated on the primary nameserver. The zone data will be transferred to the secondary nameserver via AXFR. The primary nameserver is not recorded as nameserver for the zone.
zone / main	0 - 1		Only for ns_action primary and complete
zone / main / value	1	ip	IP address of the zone (A record)
zone / main / ttl	0 - 1	int	TTL of the zone IP address
zone / nserver	1 - 7*)		
zone / nserver / name	1	domain	Host name of the nameserver
zone / nserver / ttl	0 - 1	int	TTL of the nameserver resource records
zone / soa	1		Only for ns_action: „primary“ and „complete“
zone / soa / level	0 - 1	enum	Predefined SOA settings <ul style="list-style-type: none"> 0 = soa values will be taken over 1 - 3= predefined values See below for details concerning the predefined values .
zone / soa / t tl	0 - 1	int	Default TTL of zone resource records
zone / soa / refresh	0 - 1	int	Zone refresh value
zone / soa / retry	0 - 1	int	Zone retry value

Key	Occurs	Format	Description
zone / soa / expire	0 - 1	int	Zone expire value
zone / soa / ignore	0 - 1	int	Checks, if the used SOA values are in the range that DENIC allows. <u>Available values:</u> <ul style="list-style-type: none"> • 0 = check • 1 = no check
zone / soa / email	0 - 1	email	Email address of the zone contact.
zone / source	1	string	Nameserver, the zone is copied from via AXFR (for zone_import)
zone / system_ns	0 - 1	string	The first nameserver managed by the system
zone / rr	0 - n		Only for ns_action „primary“ and „complete“
zone / rr / name	1	string	Name of the resource record
zone / rr / type	1	enum	Resource record type. The available types are: A = IPv4 address of the host AAAA = IPv6 address of the host MX = Mail Exchange – the mailserver for the domain CName = Canonic name for the host (the domain with this RR is an alias) NS = Host name of the authoritative nameserver PTR = Domain Name Pointer (for Reserve mapping to assign Names to IP addresses) TXT = free text HINFO = Host information (Type of processor and operating system) SRV = Service SPF = Sender Policy Framework NAPTR = (Naming Authority Pointer) extension of the A resource records For further information about the resource record types, see: http://en.wikipedia.org/wiki/List_of_DNS_record_types
zone / rr / pref	0 - 1	int	Resource record priority(e.g. for MX)
zone / rr / value	1	string	Resource record value
zone / rr / ttl	0 - 1	int	TTL of the resource record
zone / allow_transfer_from	0 - 1	string	IP addresses or nets, semicolon separated, semicolon at the end
zone / allow_transfer	0 - 1	enum	Allow AXFR <ul style="list-style-type: none"> • 0 = no • 1 = yes
zone / www_include	0 - 1	enum	Automatic generation of resource records „www“ <ul style="list-style-type: none"> • 0 = no • 1 = yes • Default= 1
zone / free	0 - n	string	Free records for the DNS.

Key	Occurs	Format	Description
			<p>ATTENTION !! Please only use this option if you are knowledgeable about DNS, since a wrong entry can cause the non-accessibility of your domain.</p>
default / rr_add			Add zone entries to an existing zone
default / rr_add / name	1	string	Name of the subdomain you want to change the values for. (Zone Update)
default / rr_add / ttl	0 - 1	int	TTL of the zone IP address
default / rr_add / type	1	enum	<p>Resource record type.</p> <p>The available types are:</p> <p>A = IPv4 address of the host AAAA = IPv6 address of the host MX = Mail Exchange – the mailserver for the domain CName = Canonic name for the host (the domain with this RR is an alias) NS = Host name of the authoritative nameserver PTR = Domain Name Pointer (for Reserve mapping to assign Names to IP addresses) TXT = free text HINFO = Host information (Type of processor and operating system) SRV = Service SPF = Sender Policy Framework NAPTR = (Naming Authority Pointer) extension of the A resource records</p> <p>For further information about the resource record types, see: http://en.wikipedia.org/wiki/List_of_DNS_record_types</p>
default / rr_add / pref	0 - 1	int	Resource record priority(e.g. for MX)
default / rr_add / value	1	ip	IP address of the zone (A record)
default / rr_rem			Delete zone entries inexisting zones
default / rr_rem / name	1	string	(Zone Update)Name of the subdomain you want to delete the values for.
default / rr_rem / ttl	0 - 1	int	TTL of the resource record
default / rr_rem / type	1	enum	<p>Resource record type. The available types are:</p> <p>A = IPv4 address of the host AAAA = IPv6 address of the host MX = Mail Exchange – the mailserver for the domain CName = Canonic name for the host (the domain with this RR is an alias) NS = Host name of the authoritative nameserver PTR = Domain Name Pointer (for Reserve mapping to assign Names to IP addresses) TXT = free text</p>

Key	Occurs	Format	Description
			HINFO = Host information (Type of processor and operating system) SRV = Service SPF = Sender Policy Framework NAPTR = (Naming Authority Pointer) extension of the A resource records For further information about the resource record types, see: http://en.wikipedia.org/wiki/List_of_DNS_record_types
default / rr_rem / pref	0 - 1	int	Resource record priority(e.g. for MX)
default / rr_rem / value	1	ip	IP address of the zone (A record)
default / search_and_replace	0 - n	string	Search and replace values in existing zones (Zone Update)
default / search_and_replace / search	0 - n	string	Value that should be searched for
default / search_and_replace / replace	0 - n	string	Value that should replace the search term
default / allow_transfer_from_add	0 - 1	ip	Add IP addresses or nets separated with ","
default / allow_transfer_from_rem	0 - 1	ip	Remove IP addresses or nets separated with ","

*) Depends on the registry.

Details on the predefined SOA levels

SOA Level	User Interface Label	Refresh	Retry	Expire	TTL
1	Recommended Settings	43200	7200	1209600	86400
2	High Reliability	43200	7200	1209600	43200
3	Fast Updates	43200	7200	1209600	600

9.2 Zone Create (0201)

With this task you can create a zone on the nameservers administrated by the system.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0201</code>
    <zone>
      <name>example.com</name>
      <main>
        <value>172.16.100.1</value>
        <ttl>86400</ttl>
      </main>
      <ns_action>complete</ns_action>
      <www_include>1</www_include>
      <allow_transfer_from>198.51.100.0/24</allow_transfer_from>
      <soa>
        <level>1</level>
        <refresh>43200</refresh>
        <retry>7200</retry>
        <expire>1209600</expire>
        <ttl>86400</ttl>
        <email>zonec@example.com</email>
      </soa>
      <nserver>
        <name> ns1.example.com</name>
        <ttl>86400</ttl>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
        <ttl>86400</ttl>
      </nserver>
      <rr>
        <name>mail</name>
        <ttl>103600</ttl>
        <type>MX</type>
        <pref>10</pref>
        <value>mail.example.com</value>
      </rr>
      <rr>
        <name>www</name>
        <ttl>86400</ttl>
        <type>NS</type>
        <pref></pref>
        <value>ns1.example.com</value>
      </rr>
      <!-- -->
      <rr>
        <name>default._domainkey</name>
        <type>TXT</type>
        <pref></pref>
        <ttl></ttl>
        <value>p=mVMdtsXfXxyakdNwK3CB/5nnV6dVSLTthZGMpKE9rLkfwIDAQAB</value>
      </rr>
      <free>internetconfiguration 1800 IN A 11.22.33.44</free>
      <free>mail 1800 IN A 22.33.44.55</free>
    </zone>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0201</code>
      <text>Zone has been stored on the nameserver.</text>
      <text></text>
      <type>success</type>
      <object>
        <type>zone</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
</response>
```

9.3 Zone Update (0202, 0202001)

With this task you can update a zone on the nameservers administered by the system.

Besides a complete domain update request you can at your choosing use the keys `rr_add` and `rr_rem` to add or remove resource records(for example subdomains) for a single zone record.

In addition to the zone update wizard in the interface we provide you with two comfortable functions for changing/updating your zone records : the addition and removal and searchand replacement of your zone records.

Add or remove values in zone records

With the help of the keys `rr_add` (`rr_add` = resource record add) and `rr_rem` (`rr_rem` = resource record remove) you can add or remove single and/or multiple zone records of a subdomain.

Search and replace values in zone records.

The key `search_and_replace` allows you to search for a specific type of record with an associated value and lets you replace it with another value. When using this key, the `rr_add` and `rr_rem` blocks are not required but can be used optionally without interfering the functionality of this key.

9.4 Zone Import (0204)

With this task you can import zones to the nameservers administered by the system.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0204</code>
    <default>
      <source>ns1.example.de</source>
      <ns_action>primary</ns_action>
      <soa>
        <email>email@example.de</email>
      </soa>
      <nserver>
        <name>ns1.example.com</name>
      </nserver>
      <nserver>
        <name>ns2.example.com</name>
      </nserver>
    </default>
    <zone>
      <name>example.com</name>
    </zone>
    <zone>
      <name>example.com</name>
    </zone>
    <reply_to>mail@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0204</code>
      <text>.Zone import has been completed successfully.</text>
      <text></text>
      <type>success</type>
    </status>
  </result>
</response>
```

9.5 Zone Delete (0203)

With this task you can delete a zone on the nameservers administered by the system.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
```

```

    <context></context>
  </auth>
  <task>
    <code>0203</code>
    <zone>
      <name>example.com</name>
      <system_ns>ns1.example.com</system_ns>
    </zone>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0203</code>
      <text>Zone was deleted successfully from the name server.</text>
      <type>success</type>
      <object>
        <type>zone</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
</response>

```

9.6 Zone Inquire Tasks (0205)

Dieser Task kann sowohl für Einzelabfragen (Zone Info) als auch für Listenabfragen (ZoneList) verwendet werden.

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
changed	Date of the last update.	yes		yes		yes	yes
created	Creation date of the zone	yes		yes		yes	yes
domainsafe	Usage of DomainSafe. Yes/true or No/false	yes		yes			
mainip	Main IP address of the domain	yes			yes	yes	yes

Value	Description	Info	Info <key>	List	List <key>	order	where
name	Domain name	yes		yes		yes	yes
nsaction	Additional nameserver check	yes					
ns_group	the used premium nameserver groups			yes		yes	yes
nserver	Nameserver name	yes				yes	yes
owner	Object owner	yes		yes		yes	yes
primary	Primary nameserver				yes	yes	yes
secondary1	First secondary nameserver				yes	yes	yes
secondary2	Second secondary nameserver				yes	yes	yes
secondary3	Third secondary nameserver				yes	yes	yes
secondary4	Forth secondary nameserver				yes	yes	yes
secondary5	Fifth secondary nameserver				yes	yes	yes
secondary6	Sixth secondary nameserver				yes	yes	yes
secondary7	Seventh secondary nameserver				yes	yes	yes
soa	SOA level	yes			yes	yes	
system_ns	The system nameserver	yes		yes		yes	yes
updated_by	Last updater	yes		yes		yes	yes
www_include	A www zone entry was created automatically	yes					
allow_transfer_from	IP addresses or nets domain transfer is allowed from	yes					
rr	Additional nameserver entries	yes					

9.7 Zone Info (0205)

With this task you can inquire the zone data on the nameservers administered by the system.

Example Request(single inquire)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0205</code>
    <zone>
      <name>example.com</name>
      <system_ns>ns1.example.com</system_ns>
    </zone>
    <key></key>
  </task>
</request>
```

Example Response (single inquire)

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <zone>
        <changed>2009-04-20 17:37:33</changed>
        <main>
          <value>128.122.1.3</value>
        </main>
        <name>example.com</name>
        <ns_action>complete</ns_action>
        <nserver>
          <name>ns1.example.com</name>
        </nserver>
        <nserver>
          <name>ns2.example.com</name>
        </nserver>
        <owner>
          <context></context>
          <user>customer</user>
        </owner>
        <soa>
          <email>zonec@example.com</email>
          <expire>1209600</expire>
          <refresh>43200</refresh>
          <retry>7200</retry>
          <ttl>86400</ttl>
        </soa>
        <updated_by>
          <context></context>
          <user>customer</user>
        </updated_by>
        <www_include>0</www_include>
      </zone>
    </data>
    <status>
      <code>S0205</code>
      <text>Zone information was inquired successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

9.8 Zone List (0205)

With this task you can inquire zone data of several zones on the nameservers administered by the system.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0205</code>
    <view>
      <offset>0</offset>
      <limit>4</limit>
      <children>1</children>
    </view>
    <key>created</key>
    <key>primary</key>
    <key>system_ns</key>
    <key>secondary2</key>
    <key>mainip</key>
    <key>changed</key>
    <where>
      <key>name</key>
      <operator>like</operator>
      <value>*</value>
    </where>
    <order>
      <key>changed</key>
      <mode>DESC</mode>
    </order>
    <order>
      <key>changed</key>
      <mode>DESC</mode>
    </order>
  </task>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>1</summary>
      <zone>
        <changed>2014-12-02 09:49:20</changed>
        <created>2013-11-12 13:37:04</created>
        <name>example.com</name>
        <primary>ns1.example.com</primary>
        <domainsafe>0</domainsafe>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updated_by>
          <user>customer</user>
          <context>1018945</context>
```

```
        </updated_by>
        <system_ns>ns1.example.com</system_ns>
    </zone>
</data>
<status>
    <code>S0205</code>
    <text>Zone information was inquired successfully.</text>
    <type>success</type>
</status>
</result>
<stid>20160218-app2-dev-2603</stid>
</response>
```


10 DNSSEC

DNSSEC (DOMAIN NAME SYSTEM SECURITY EXTENSIONS)

DNSSEC is an extension to the domain name system. The primary purpose of DNSSEC is to ensure that the user receives the data from the requested website and not a different website.

An example of this would be in the case of cache poisoning within the nameservers of an internet service provider which redirects the request and sends the user to an unwanted website.

DNSSEC takes care of the authenticity of the data and guarantees the completeness of the returned information. Furthermore, DNSSEC also guarantees that the data has not been tampered with. This process takes place by signing the DNS records in the nameservers.

10.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	DNSec Sign 0120001 DNSec Rollover 0120002 DNSec Unsign 0120003 DNSec Info 0120004 DNSec List 0120005
auto_dnssec	1 - n		
auto_dnssec / name	1	string	Name of the domain that you want to be signed

NOTE

Please take in mind that you have to use NodeSecure, our AnyCast nameserver service, if you want to use DNSSEC.

The NodeSecure name servers are distributed worldwide and guarantee an extremely high availability with short access times. In addition NodeSecure provides the ability to sign zones with DNSSEC without additional costs. Please contact our sales department for comprehensive information about this product.

IMPORTANT NOTES

- Domains, that are signed with DNSSEC can not be deleted. The domains must first be unsigned successfully.
- Zones for which the domains are signed with DNSSEC cannot be deleted. The domains must first be unsigned.
- An update to a different nameserver cannot take place as long as the signing or unsigned procedure is running and not completed.
- The signing and the key rollover of a domain can take up to 2 days, the unsigned up to 10 days.

10.2 DNSSec Sign (0120001)

With this task you can sign domains with DNSSEC.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <context></context>
    <password>password</password>
  </auth>
  <owner>
    <user>customer2</user>
    <context></context>
  </owner>
  <task>
    <code>0120001</code>
    <auto_dnssec>
      <name>example.com</name>
    </auto_dnssec>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <auto_dnssec_job>
        <job>
          <id>3582765</id>
          <status>NOT_SET</status>
        </job>
      </auto_dnssec_job>
    </data>
    <status>
      <code>N0120001</code>
      <text>The DNSSEC signing was started successfully.</text>
      <type>notify</type>
      <object>
        <type>auto_dnssec</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
  <stid>20150630-app2-qm-1214</stid>
</response>
```

10.3 DNSSec Key-Rollover (0120002)

With this task you renew the KSK key (key signing key) of a DNSSEC signed domain. The KSK is transferred automatically to the registry. A key rollover can be required due to operational or security reasons when one or more keys have been compromised.

WHAT IS A KSK?

The Key Signing Key (KSK) / DNSKEY is the key that is used to sign the ZSK. The Key Signing Key is sent from the registrar to the registry.

By using the KSK, the authenticity of the data that is sent from the TLD rootserver to the provider's nameserver is guaranteed while it is being handed off to the next respective instance in the chain of trust.

WHAT IS A ZSK?

ZSK means Zone Signing Key. The Zone Signing Key is the key which is used to sign entries within a zone. The ZSK is stored on the nameserver for each zone.

Examples for XML Requests and Answers**Example Request**

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <context></context>
    <password>password</password>
  </auth>
  <owner>
    <user>customer2</user>
    <context></context>
  <task>
    <code>0120002</code>
    <auto_dnssec>
      <name>example.com</name>
    </auto_dnssec>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <auto_dnssec_job>
        <job>
          <id>3603918</id>
          <status>NOT_SET</status>
        </job>
      </auto_dnssec_job>
    </data>
    <status>
      <code>N0120002</code>
      The DNSSEC key rollover was startet successfully.</text>
      <type>notify</type>
      <object>
        <type>auto_dnssec</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
  <stid>20150702-app1-qm-282</stid>
</response>
```

10.4 DNSSec Unsign (0120003)

With this task you remove the DNSSEC-signing of a domain.

NOTE

Also applicable for domains which are managed with external nameservers

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <context></context>
    <password>password</password>
  </auth>
  <owner>
    <user>customer2<</user>
    <context></context>
  </owner>
  <task>
    <code>0120003</code>
    <auto_dnssec>
      <name>example.com</name>
    </auto_dnssec>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <auto_dnssec_job>
        <job>
          <id>3582970</id>
          <status>NOT_SET</status>
        </job>
      </auto_dnssec_job>
    </data>
    <status>
      <code>N0120003</code>
      <text>The request for deletion of the DNSSEC signing was startet.</text>
      <type>notify</type>
      <object>
        <type>auto_dnssec</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
  <stid>20150630-app1-qm-989</stid>
</response>
```

10.5 DNSSec Inquire Tasks (0120004, 0120005)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
created	Signing date	yes		yes		yes	yes
dnssec	Usage of DNSSEC	yes					
dns_record_remo ved	Timestamp displaying when the record was removed from the parent zone.	yes			yes		
idn	Original writing of the domain name	yes		yes		yes	yes
last_rollover	Date of the last executed key-rollover	yes		yes		yes	yes
name	Name of the signed domain	yes		yes		yes	yes
owner	Object owner	yes		yes		yes	yes
owner / context	Owner of the domain object						
owner / user	User name of the object owner						
updated	Last updated	yes		yes		yes	yes
updater	User who has made the last update	yes			yes	yes	yes
updater /user	Name of the user who made the last update.						
updater /context	Context of the user who made the last update.						

10.6 DNSSec Info (0120004)

With this task you can inquire the last key rollover of a DNSSEC signed domain.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <context></context>
    <password>password</password>
  </auth>
  <owner>
    <user>customer2</user>
    <context></context>
    <task>
      <code>0120004</code>
      <auto_dnssec>
        <name>example.com</name>
      </auto_dnssec>
    </task>
  </request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <auto_dnssec>
        <name>example.com</name>
        <owner>
          <user>customer2</user>
          <context></context>
        </owner>
        <updater>
          <user>customer2</user>
          <context></context>
        </updater>
        <created>2015-07-02 09:52:13</created>
        <updated>2015-07-02 09:52:13</updated>
      </auto_dnssec>
    </data>
    <status>
      <code>S0120004</code>
      <text>DNSSEC-Signierung wurde erfolgreich ermittelt.</text>
      <type>success</type>
      <object>
        <type>auto_dnssec</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
  <stid>20150702-app2-qm-79</stid>
</response>
```

10.7 DNSSec List (0120005)

With this task you can inquire the last key rollover of DNSSEC signed domains.

The values of these keys will be returned by default:

name, idn, created, updated, last_rollover (Datum der letzten Ausführung eines Key-Rollover)

Following data can be inquired additionally by a list inquire using the key key:

owner, ds_record_removed

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <context></context>
    <password>password</password>
  </auth>
  <owner>
    <user>customer2<</user>
    <context></context>
  </owner>
  <task>
    <code>0120005</code>
    <view>
      <children>1</children>
    </view>
  </task>
</request>
```

Example Response

```
?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>2</summary>
      <auto_dnssec>
        <name>example.com</name>
        <last_rollover>2015-06-30 17:10:40</last_rollover>
        <created>2015-06-30 16:54:55</created>
        <updated>2015-06-30 17:21:01</updated>
      </auto_dnssec>
      <auto_dnssec>
        <name>example2.com</name>
        <created>2015-07-02 09:52:13</created>
        <updated>2015-07-02 09:52:13</updated>
      </auto_dnssec>
    </data>
    <status>
      <code>S0120005</code>
      <text>DNSSEC-Signierungen wurden erfolgreich ermittelt.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20150702-app1-qm-174</stid>
</response>
```

11 BackupMX

A **BackupMX** entry specifies a backup mail server to store your mails in case the primary mail server breaks down and is unreachable. When the primary mail server is operational again, the BackupMX server automatically sends the stored mails to your primary mail server.

11.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	BackupMX Create 0521 BackupMX Delete 0523 BackupMX Inquire 0524
backup_mx	1-n		
backup_mx / domain	1	string	Domain name

11.2 BackupMX Create (0521)

With this task you can create new BackupMX entries.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <!--
  <owner>
    <user>subuser</user>
    <context></context>
  </owner>
  -->
  <task>
    <code>0521</code>
    <backup_mx>
      <domain>example2.de</domain>
    </backup_mx>
    <backup_mx>
      <domain>example2.com</domain>
    </backup_mx>
    <reply_to>customer@example.com</reply_to>
    <ctid>customer-transaction-id</ctid>
  </task>
</request>
```


Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <msg>
      <text>BackupMX wurde erfolgreich eingetragen.</text>
      <code>S0521</code>
      <type>success</type>
      <object>
        <type>backup_mx</type>
        <value>example2.de</value>
      </object>
    </msg>
    <msg>
      <text>BackupMX successfully registered.</text>
      <code>S0521</code>
      <type>success</type>
      <object>
        <type>backup_mx</type>
        <value>example2.com</value>
      </object>
    </msg>
    <status>
      <code>S0521</code>
      <text>BackupMX successfully registered.</text>
      <type>success</type>
    </status>
  </result>
  <ctid>customer-transaction-id</ctid>
  <stid>20150429-appl-test-5118</stid>
</response>
```

11.3 BackupMX Inquire (0524)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
created	Date, the entry was created	yes		yes		yes	yes
domain	Domain name	yes		yes		yes	yes

Value	Description	Info	Info <key>	List	List <key>	order	where
owner	Object owner	yes					
updated	Last updated	yes					
updater	User, who made the last update	yes					

11.4 BackupMX Info (0524)

With this task you can inquire the data of a BackupMX entry.

Examples for XML Requests and Answers

Example Request Info

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <!--
  <owner>
    <user>subuser</user>
    <context></context>
  </owner>
  -->
  <task>
    <code>0524</code>
    <backup_mx>
      <domain>example.com</domain>
    </backup_mx>
    <reply_to>cutomer@example.com</reply_to>
    <ctid>customer-transaction-id</ctid>
  </task>
</request>
```

Example Response Info

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <backup_mx>
        <domain>example.com</domain>
        <created>2010-10-13 16:21:40</created>
        <updated>2010-10-13 16:21:40</updated>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updater>
          <user>customer</user>
          <context>1018945</context>
        </updater>
      </backup_mx>
    </data>
    <status>
      <code>S0524</code>
```

```

    <text>BackupMX data inquired successfully.</text>
    <type>success</type>
    <object>
      <type>backup_mx</type>
      <value>example.com</value>
    </object>
  </status>
</result>
<ctid>customer-transaction-id</ctid>
<stid>20150429-app1-test-5031</stid>
</response>

```

11.5 BackupMX List (0524)

With this task you can inquire data of several BackupMX entries.

Examples for XML Requests and Answers

Example RequestList

```

<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <!--
  <owner>
    <user>subuser</user>
    <context>4</context>
  </owner>
  -->
  <task>
    <code>0524</code>
    <view>
      <limit>3</limit>
      <children>10</children>
    </view>
    <where>
      <key>domain</key>
      <operator>like</operator>
      <value>*</value>
    </where>
    <reply_to>customer@example.com</reply_to>
    <ctid>customer-transaction-id</ctid>
  </task>
</request>

```

Example ResponseList

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>2</summary>
      <backup_mx>
        <domain>example2.com</domain>
        <created>2015-04-29 10:57:13</created>
      </backup_mx>
      <backup_mx>
        <domain>example2.de</domain>
        <created>2015-04-29 10:57:13</created>

```

```

        </backup_mx>
        <backup_mx>
            <domain>example3.de</domain>
            <created>2015-04-29 10:57:13</created>
        </backup_mx>
    </data>
    <status>
        <code>S0524</code>
        <text>BackupMX data inquired successfully.</text>
        <type>success</type>
    </status>
</result>
<ctid>customer-transaction-id</ctid>
<stid>20150429-appl-test-5333</stid>
</response>

```

11.6 BackupMX Delete (0523)

With this task you can delete BackupMX entries.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0523</code>
    <backup_mx>
      <domain>example.com</domain>
    </backup_mx>
    <reply_to>customer@example.com</reply_to>
    <ctid>1234567</ctid>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data/>
    <status>
      <code>S0523</code>
      <text>BackupMX deleted successfully.</text>
      <type>success</type>
      <object>
        <type>backup_mx</type>
        <value>example.com</value>
      </object>
    </status>
  </result>
  <ctid>customer-transaction-id</ctid>
  <stid>20150429-appl-test-5071</stid>
</response>

```

12 MailProxy

The MailProxy receives the mail sent to the domain it is configured for. It then checks the mail for spam, viruses and other malware. Contaminated mail can be deleted or removed to a quarantine folder. All mail, including the marked spam, is then redirected to the target mail server.

Be aware, that the MailProxy function has to be activated.

12.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	MailProxy Create 0511 MailProxy Update 0512 MailProxy Delete 0513 MailProxy Inquire 0514
mail_proxy			
mail_proxy/ domain	1	string	Domain(s), the MailProxy is used for.
mail_proxy/ admin	0 - 1	string	Email address of the mail administrator who receives the email notifications.
mail_proxy/ target	1	string	The hostname of the mail server the mails proxy redirects the mails to.
			NOTE The MX entry for your mail server has to be deleted in the zone.
mail_proxy/ protection	0 - 1	enum	<p>Security settings fo the handling of infected mails. You can select one of three protection levels:</p> <ul style="list-style-type: none"> • high: Infected mails are moved to a quarantine folder. The subject of spam mails is modified. The string "+++SPAM" is added. • medium: Infected mails are zipped and attached to a mail that is redirected to the primary target address. The subject of spam mails is modified. The string "+++SPAM" is added. • low: Infected mails are zipped and attached to a mail that is redirected to the primary target address. <p>Banned files are treated in the same way in all three protection levels. They are zipped and attached to a mail that is redirected to the primary target address.</p>

Key	Occurs	Format	Description
			<p>NOTE If you choose this security level, other the settings that have been made will be ignored</p> <ul style="list-style-type: none"> ▪ custom = user defined. You can specify all settings yourself. ▪ off= safety settings are deactivated
mail_proxy/ greylisting	0 - 1		Settings for the greylisting feature.
mail_proxy/ greylisting / mode	0 - 1	enum	<p>Enabeling or disabling greylisting. When using greylisting, the first email from an unknown sender is rejected. Only subsequent emails from the sender are accepted.</p> <p>Settings:</p> <ul style="list-style-type: none"> ▪ enabled ▪ disabled <p>NOTE Please note that greylisting can cause the delayed delivery of mails.</p>
mail_proxy / virus	0 - 1		<p>Treating of viruses. When a virus is detected, the infected mail is zipped and attached to a mail. This mail is redirected to the primary target address containing a warning.</p>
mail_proxy / virus / mode	0 - 1	enum	<p>Specifying the treating of emails that are infected with a virus.</p> <p>Settings:</p> <ul style="list-style-type: none"> • quarantine: the filtered mails are moved to a quarantine folder. An information mail is sent daily to the administrator email address you have entered. It informs about all emails in the quarantine folder. Using a link, you can open the quarantine folder and decide how to treat each mail. • disabled: virus protection is not active. • accept: the filtered emails are accepted, even though they are filtered out. • discard: the emails will be deleted.
mail_proxy / banned_files	0 - 1		<p>Settings for banned files protection:</p> <p>Banned files are a list of file types, known to be infected with viruses often, e.g. .scr, .exe.jpg. A list with the respective file types is stored in the system.</p>

Key	Occurs	Format	Description
			<p>These file types are stored in the list: Double file extensions containing the following extensions: exe vbs pif scr bat cmd com cpl dll these MIME Types: application/x-msdownload,application/x-msdos- program,application/hta,message/partial,message/external- body</p> <p>Simple file extensions: ade adp app bas bat chm cmd com cpl crt exe fpx hlp hta inf ins isp js jse lnk mda mdb mde mdw mdt mdz msc msi msp mst ops pcd pif prg reg scr sct shb shs vb vbe vbs wsc wsf wsh these types: exe-ms, exe lha cab dll</p>
mail_proxy / banned_files /mode	0 - 1		<p>Treatment of emails containing banned files.</p> <p>Settings:</p> <ul style="list-style-type: none"> quarantine: the filtered mails are moved to a quarantine folder. An information mail is sent daily to the administrator email address you have entered. It informs about all emails in the quarantine folder. Using a link, you can open the quarantine folder and decide how to treat each mail. disabled: virus protection is not active. accept: the filtered emails are accepted, even though they are filtered out. discard: the emails will be deleted.
mail_proxy / header	0 - 1		Marking the mails as spam in the header (Header X-Spam-Status und X-Spam-Level)
mail_proxy / header / mode	0 - 1		<p>Treating of emails, which are marked as Spam in the header.</p> <ul style="list-style-type: none"> quarantine: the filtered mails are moved to a quarantine folder. An information mail is sent daily to the administrator email address you have entered. It informs about all emails in the quarantine folder. Using a link, you can open the quarantine folder and decide how to treat each mail. disabled: virus protection is not active. accept: the filtered emails are accepted, even though they are filtered out. discard: the emails will be deleted.
mail_proxy / spam /	0 - 1		Emails can be flagged as spam by the string "****SPAM" in the email subject.

Key	Occurs	Format	Description
mail_proxy / spam / modify_subject	0 - 1	enum	Setting for the email subject: 1 = the string "****SPAM" will be added to the email subject 0 = the string "****SPAM" will not be added to the email subject.
mail_proxy / spam / tag_header	0 - 1		Enter a score here. If an incoming email reaches this value, a tag that flags the mail as spam is added to the header (Header X-Spam-Status and X-Spam-Level). Recommended: values between -999 and 1.
mail_proxy / spam / spam	0 - 1		Enter a score here. If an incoming email reaches this value, it is marked as spam (X-Spam-Flag). The score is added to the header. Recommended: Values between 4.5 and 8. The lower the value, the more mails are marked as spam.
mail_proxy / spam / kill	0 - 1		Enter a score here. If an incoming email reaches this value, it is treated according to the protection settings you defined in the section "Spam Filter" and is sent to the quarantine folder when this is set. NOTE Please note that the kill score value has to be higher than the spam score value.
mail_proxy / spam / mode	0 - 1		SPAM filter settings: <ul style="list-style-type: none"> quarantine: the filtered mails are moved to a quarantine folder. An information mail is sent daily to the administrator email address you have entered. It informs about all emails in the quarantine folder. Using a link, you can open the quarantine folder and decide how to treat each mail. disabled: virus protection is not active. accept: the filtered emails are accepted, even though they are filtered out. discard: the emails will be deleted.
mail_proxy / whitelist	0 - 1		Definition of email addresses for which incoming emails should <u>never</u> be marked as spam.
mail_proxy / whitelist / email	0 - n		Add a specific email address here. Incoming mails from this email address will not be marked as spam. To include all emails sent to a domain in the whitelist, enter only @ and the domain name, e.g. "@example.com".
mail_proxy / blacklist	0 - 1		Definition of email addresses for which incoming emails should <u>always</u> be marked as spam.
mail_proxy / blacklist / email	0 - n		Add specific email addresses to the blacklist here. To include all emails sent to a domain in one of the lists, enter only @ and the domain name, e.g. "@example.com".
mail_proxy / excludelist	0 - 1		Certain administrative email addresses may not be checked by spam filters , e.g. "hostmaster@example.com".

Key	Occurs	Format	Description
mail_proxy / excludelist / email	0 - n		Add the email addresses to the exclusion list here.
			NOTE Please be aware that you may not enter the Quarantine Mail Address here.

12.2 Mailproxy Create (0511)

With this task you can create a new MailProxy entry.

Examples for XML Requests and Answers

Example Request

```
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>0511</code>
    <mail_proxy>
      <target>example1695.com</target>
      <protection>CUSTOM</protection>
      <!-- HIGH|MEDIUM|LOW|OFF|CUSTOM -->
      <greylisting>
        <mode>DISABLED</mode>
        <!-- DISABLED|ENABLED -->
      </greylisting>
      <virus>
        <mode>ACCEPT</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
      </virus>
      <banned_files>
        <mode>DISABLED</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
      </banned_files>
      <header>
        <mode>QUARANTINE</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
      </header>
      <spam>
        <modify_subject>1</modify_subject>
        <!-- 0|1-->
        <tag_header>5</tag_header>
        <!-- -999.9:999.9-->
        <spam>6</spam>
        <!-- -999.9:999.9-->
        <kill>10</kill>
        <!-- -999.9:999.9-->
        <mode>QUARANTINE</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
      </spam>
      <whitelist>
        <email>nospam@example2.com</email>
        <email>nospam2@example3.com</email>
      </whitelist>
    </mail_proxy>
  </task>
</request>
```

```

        </whitelist>
        <blacklist>
            <email>spammer@example4.com</email>
            <email>spammer2@example5.com</email>
        </blacklist>
        <excludelist>
            <email>postmaster@example6.com</email>
        </excludelist>
        <domain>example1695.com</domain>
        <admin>customer@example.com</admin>
    </mail_proxy>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data/>
    <status>
      <code>S0511</code>
      <text>MailProxy registered successfully.</text>
      <type>success</type>
      <object>
        <type>mailproxy</type>
        <value>example1695.com</value>
      </object>
    </status>
  </result>
  <stid>20150427-appl-test-5772</stid>
</response>

```

12.3 MailProxy Update (0512)

With this task you can actualize a MailProxy entry.

NOTE

Please note, that you cannot change the domain with this task.

Examples for XML Requests and Answers

Example Request

```

<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>0511</code>
    <mail_proxy>
      <target>example1695.com</target>
      <protection>CUSTOM</protection>
      <!-- HIGH|MEDIUM|LOW|OFF|CUSTOM -->
      <greylisting>
        <mode>DISABLED</mode>
        <!-- DISABLED|ENABLED -->
      </greylisting>
      <virus>

```

```

        <mode>ACCEPT</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
    </virus>
    <banned_files>
        <mode>DISABLED</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
    </banned_files>
    <header>
        <mode>QUARANTINE</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
    </header>
    <spam>
        <modify_subject>1</modify_subject>
        <!-- 0|1-->
        <tag_header>5</tag_header>
        <!-- -999.9:999.9-->
        <spam>6</spam>
        <!-- -999.9:999.9-->
        <kill>10</kill>
        <!-- -999.9:999.9-->
        <mode>QUARANTINE</mode>
        <!-- QUARANTINE|DISABLED|ACCEPT|DISCARD -->
    </spam>
    <whitelist>
        <email>nospam@example2.com</email>
        <email>nospam2@example3.com</email>
    </whitelist>
    <blacklist>
        <email>spammer@example4.com</email>
        <email>spammer2@example5.com</email>
    </blacklist>
    <excludelist>
        <email>postmaster@example6.com</email>
    </excludelist>
    <domain>example1695.com</domain>
    <admin>customer@example.com</admin>
    </mail_proxy>
</task>
</request

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data/>
    <status>
      <code>S0512</code>
      <text>MailProxy updated successfully.</text>
      <type>success</type>
      <object>
        <type>mailproxy</type>
        <value>example1695.com</value>
      </object>
    </status>
  </result>
  <ctid/>
  <stid>20150427-app1-test-6059</stid>
</response>

```

12.4 MailProxy Inquire (0514)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
admin	Email address of the mail administrator who receives the email notifications.	yes					
protection	Security settings fo the handling of infected mails	yes		yes		yes	yes
domain	Domain(s), the MailProxy is used for.	yes		yes		yes	yes
created	Creation date of the MailProxy	yes		yes		yes	yes
target	The hostname of the mail server the Mailproxy redirects the mails to.		yes				
owner	Object owner		yes				
greylisting	Settings for the greylisting feature.	yes					
virus	Treating of viruses	yes					
banned files	Settings for banned files protection:	yes					
header	Treating of emails, which are markedd as Spam in the header	yes					
spam	Emails can be flagged as spam by the string "****SPAM" in the email subject.	yes					
whitelist	Definition of email addresses for which incoming emails should <u>never</u> be marked as spam.	yes					
blacklist	Definition of email addresses for which incoming emails should <u>always</u> be marked as spam.	yes					
updated	Last updated	yes					
updater	User who made the last update	yes					

12.5 Mailproxy Info (0514)

With this task you can inquire the MailProxy data of one (info) or several (list) domains.

Examples for XML Requests and Answers

Example Request (Info)

```
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <!--
  <owner>
    <user>subuser</user>
    <context>104</context>
  </owner>
  -->
  <task>
    <code>0514</code>
    <mail_proxy>
      <domain>example1695.com</domain>
    </mail_proxy>
    <reply_to>customer@example.com</reply_to>
    <ctid>customer-transaction-id</ctid>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <mail_proxy>
        <target>example1695.com</target>
        <admin>customer@example.com</admin>
        <protection>CUSTOM</protection>
        <greylisting>
          <mode>DISABLED</mode>
        </greylisting>
        <virus>
          <mode>ACCEPT</mode>
        </virus>
        <banned_files>
          <mode>DISABLED</mode>
        </banned_files>
        <header>
          <mode>QUARANTINE</mode>
        </header>
        <spam>
          <modify_subject>1</modify_subject>
          <tag_header>5.0</tag_header>
          <spam>6.0</spam>
          <kill>10.0</kill>
          <quarantine_digest_interval>1440</quarantine_digest_interval>
          <mode>QUARANTINE</mode>
        </spam>
        <whitelist>
          <email>nospam2@example3.com</email>
          <email>nospam@example2.com</email>
        </whitelist>
        <blacklist>
          <email>spammer2@example5.com</email>
        </blacklist>
      </mail_proxy>
    </data>
  </result>
</response>
```

```

        <email>spammer@example4.com</email>
    </blacklist>
    <domain>example1695.com</domain>
    <created>2015-04-27 13:40:32</created>
    <updated>2015-04-27 13:40:32</updated>
    <owner>
        <user>customer</user>
        <context></context>
    </owner>
    <updater>
        <user>customer</user>
        <context>1018945</context>
    </updater>
</mail_proxy>
</data>
<status>
    <code>S0514</code>
    <text>MailProxy data inquired successfully.</text>
    <type>success</type>
    <object>
        <type>mailproxy</type>
        <value>example1695.com</value>
    </object>
</status>
</result>
<ctid>customer-transaction-id</ctid>
<stid>20150427-app1-test-6217</stid>
</response>

```

Example Request (List)

```

<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <!--
  <owner>
    <user>subuser</user>
    <context>104</context>
  </owner>
  -->
  <task>
    <code>0514</code>
    <view>
      <limit>2</limit>
      <children>1</children>
    </view>
    <where>
      <key>domain</key>
      <operator>like</operator>
      <value>ex*</value>
    </where>
    <reply_to>customer@example.com</reply_to>
    <ctid>customer-transaction-id</ctid>
  </task>
</request>

```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data>
      <summary>3</summary>
      <mail_proxy>
        <admin>customer@example123.de</admin>
        <protection>CUSTOM</protection>
        <domain>example123.de</domain>
        <created>2011-12-01 11:17:09</created>
      </mail_proxy>
      <mail_proxy>
        <admin>customer@example456.de</admin>
        <protection>MEDIUM</protection>
        <domain>example456.de</domain>
        <created>2015-11-04 11:27:55</created>
      </mail_proxy>
      <mail_proxy>
        <admin>customer@example789.de</admin>
        <protection>CUSTOM</protection>
        <domain>example789.de</domain>
        <created>2015-12-03 14:53:55</created>
      </mail_proxy>
    </data>
    <status>
      <code>S0514</code>
      <text>MailProxy data inquired successfully.</text>
      <type>success</type>
    </status>
  </result>
  <ctid>customer-transaction-id</ctid>
  <stid>20160412-app2-359428</stid>
</response>
```

12.6 MailProxy Delete (0513)

With this task you can delete a MailProxy entry.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>0513</code>
    <mail_proxy>
      <domain>example.de</domain>
    </mail_proxy>
    <!-- This task is bulk-capable and you may remove multiple records -->
    <!--
    <mail_proxy>
      <domain>example2.com</domain>
    </mail_proxy>
    <mail_proxy>
      <domain>example3.com</domain>
```

```
</mail_proxy>
-->
<reply_to>EMAIL</reply_to>
<ctid>customer-transaction-id</ctid>
</task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <msg>
      <text>MailProxy deleted successfully.</text>
      <code>S0513</code>
      <type>success</type>
      <object>
        <type>mailproxy</type>
        <value>example165.com</value>
      </msg>
      <object>
        <type>mailproxy</type>
        <value>example1695.de</value>
      </object>
    </msg>
    <msg>
      <text>MailProxy deleted successfully.</text>
      <code>S0513</code>
      <type>success</type>
      <object>
        <type>mailproxy</type>
        <value>beispiel-24.net</value>
      </object>
    </msg>
    <status>
      <code>N0513</code>
      <text>MailProxy request has been processed successfully.</text>
      <type>notify</type>
    </status>
  </result>
  <ctid>customer-transaction-id</ctid>
  <stid>20150427-app1-test-6478</stid>
</response>
```


13 Domain Contacts (Handles)

13.1 Description of the XML Keywords

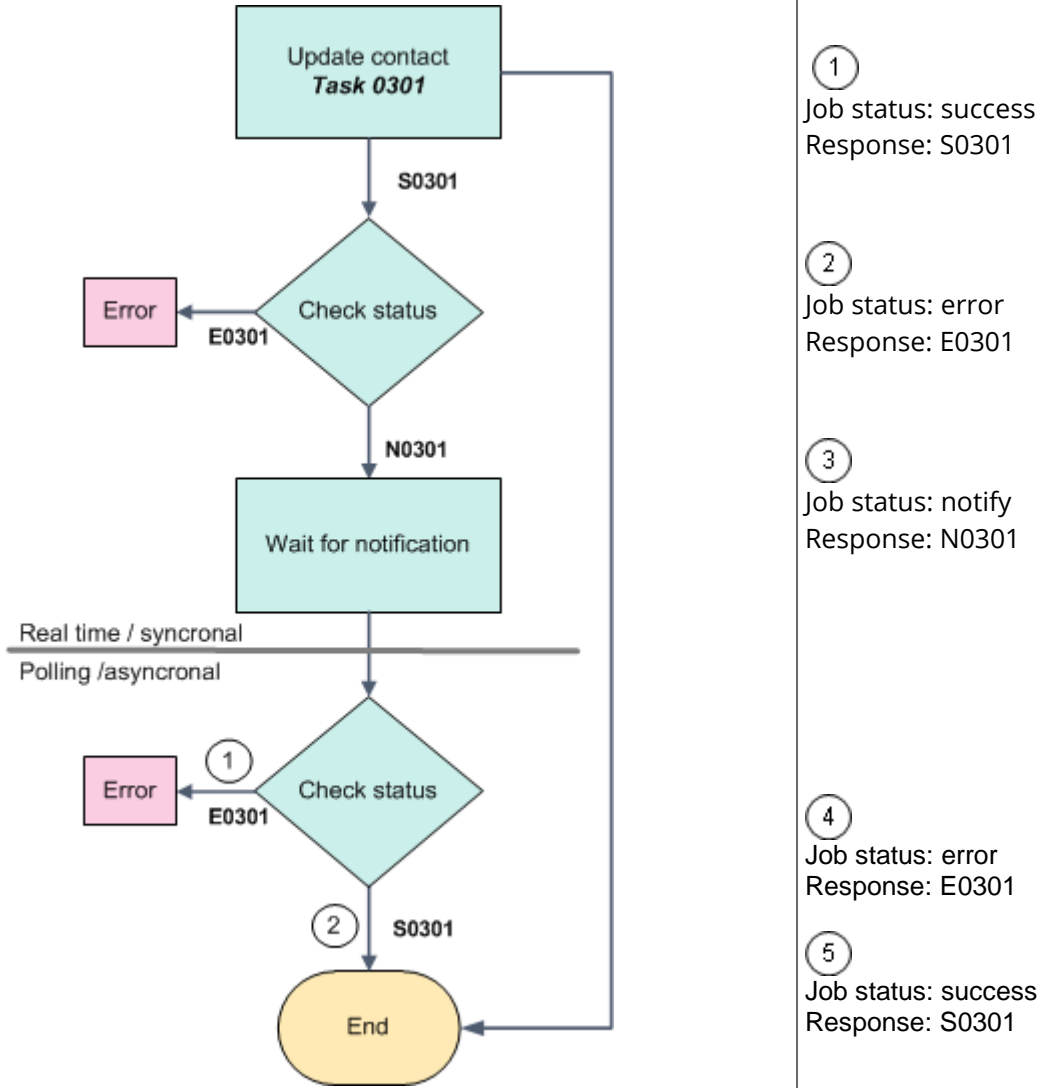
XML Key	Occurs	Format	Description
code	1	task	Contact Create 0301 Contact Update 0302 Contact Delete 0303 Contact Inquire 0304
handle / type	1	enum	Domain contact type <ul style="list-style-type: none"> PERSON ORG ROLE
handle / fname	1	string	First name
handle / lname	1	string	Last name
handle / title	0 - 1	string	Title
handle / organization	1	string	Organization
handle / address	1 - 3	string	Street, it depends on the registry how many characters are allowed.
handle / pcode	1	string	Postal code
handle / city	1	string	City (place of residence)
handle / state	0 - 1	string	State
handle / country	1	country	Country (ISO 3166 Country Code)
handle / phone	0 - 1*)	phone	Telephone number e.g.:+1-123-123
handle / fax	0 - 1*)	phone	Fax number e.g.:+1-123-123
handle / email	0 - 1*)	email	Email address
handle / sip	0 - 1*)	URI	Sip address
handle / protection	1	enum	Data, to be displayed in a Whois request <ul style="list-style-type: none"> a = all b = (Default) some data is not displayed (Depends on the registry and the usage of the domain contacts)
nic_ref	0 - n	country	Generates a NIC reference for the domain contact
remarks	0 - 1	string	Remarks
reply_to	1	string	Email address to which the reply is send
handle/ extension / mode	0 - 1	enum	This key specifies how contact extensions are treated when updating a contact.

XML Key	Occurs	Format	Description
			<ul style="list-style-type: none">merge =all existing keys remain, if the value is different, it will be updated. New keys are added.replace =existing keys are deleted, the new ones are set. Default is "merge"
handle / force_handle_create	0 -1	enum	1 = No check for duplicate contacts. The new contact is always created even if the same contact already exists.
handle /comment	0 - 1	string	Domain contact comment. Only ASCII characters allowed.
handle / id	0 -1	string	Domain contact ID

*) Depends on the registry.

13.2 Contact Create (0301)

With this task you can create a new domain contact (handle).



Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0301</code>
    <handle>
      <type>PERSON</type>
      <fname>Peter</fname>
      <lname>Doe</lname>
      <title></title>
      <organization>PDA</organization>
      <address>Example Street 3</address>
      <pcode>12345</pcode>
    </handle>
  </task>
</request>
```

```

<city>Example City</city>
<state></state>
<country>USA</country>
<phone>+49-12345-12345</phone>
<fax>+49-12345-12345</fax>
<email>Doe@example.com</email>
<sip></sip>
<protection>B</protection>
<nic_ref>de</nic_ref>
<nic_ref>com</nic_ref>
<nic_ref>net</nic_ref>
<remarks></remarks>
<extension>
  <gender></gender>
  <idnumber></idnumber>
  <vatnumber></vatnumber>
  <trademark_name></trademark_name>
  <trademark_number></trademark_number>
  <trademark_country></trademark_country>
  <trademark_regdate></trademark_regdate>
  <trademark_appdate></trademark_appdate>
  <companynumber></companynumber>
  <birthday></birthday>
  <birthplace></birthplace>
  <birthcountry></birthcountry>
</extension>
</handle>      <force_handle_create>1</force_handle_create>
<reply_to>customer@example.com</reply_to>
</task>
</request>

```

Example Response

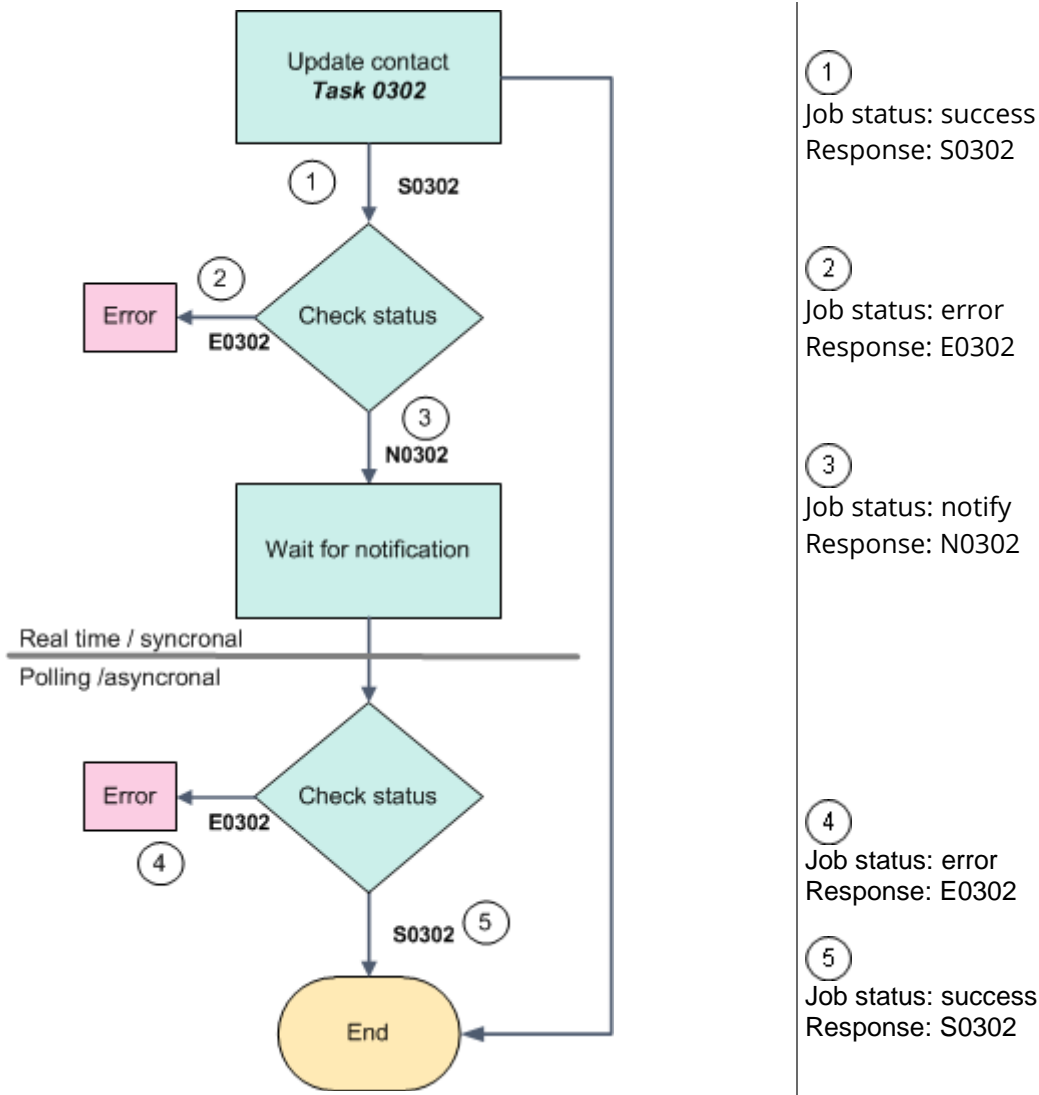
```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0301</code>
      <text>Domain contact request has been started.</text>
      <text></text>
      <type>notify</type>
      <object>
        <type>handle</type>
        <value>9926612</value>
      </object>
    </status>
  </result>
</response>

```

13.3 Contact Update (0302)

With this task you can actualize the domain contact data.



WARNING!!

Domain contact data, which is not sent in the domain contact update request will be deleted!

Example:

If you send a domain contact update request without a fax number, the fax number will be deleted by the system. To be sure that no data will be deleted, always send all data to the system.

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  
```

```

</auth>
<task>
  <code>0302</code>
  <handle>
    <alias></alias>
    <type>PERSON</type>
    <fname>Peter</fname>
    <lname>Doe</lname>
    <title></title>
    <organization>PDA</organization>
    <address>Example Street 3</address>
    <pcode>12345</pcode>
    <city>Example City</city>
    <state></state>
    <country>USA</country>
    <phone>+49+1-12345-12345</phone>
    <fax>+49+1-12345-12345</fax>
    <email>Doe@example.com</email>
    <sip></sip>
    <protection>B</protection>
    <nic_ref>de</nic_ref>
    <nic_ref>com</nic_ref>
    <nic_ref>net</nic_ref>
    <remarks></remarks>
    <extension>
      <gender></gender>
      <idnumber></idnumber>
      <vatnumber></vatnumber>
      <trademark_name></trademark_name>
      <trademark_number></trademark_number>
      <trademark_country></trademark_country>
      <trademark_regdate></trademark_regdate>
      <trademark_appdate></trademark_appdate>
      <companynumber></companynumber>
      <birthday></birthday>
      <birthplace></birthplace>
      <birthcountry></birthcountry>
    </extension>
  </handle>
  <force_handle_create>1</force_handle_create>
  <reply_to>customer@example.com</reply_to>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>N0302</code>
      <text>Update of domain contact has been started.</text>
      <text></text>
      <type>notify</type>
      <object>
        <type>handle</type>
        <value>9926612</value>
      </object>
    </status>
  </result>
</response>

```

13.4 Contact Comment Update (0302004)

With this task you can create and update a domain contact comment. The <default>-bloc ist only required for bulk requests.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0302004</code>
    <handle>
      <id>1234567</id>
      <comment>This is another comment.</comment>
    </handle>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data/>
    <status>
      <code>S0302004</code>
      <text>Domain contact comment was changed successfully.</text>
      <type>success</type>
      <object>
        <type>handle</type>
        <value>1234567</value>
      </object>
    </status>
  </result>
  <stid>20150120-app1-test-5928</stid>
</response>
```

13.5 Contact Delete (0303)

With this task you can delete domain contacts (handles).

NOTE

A domain contact which is still assigned to a domain cannot be deleted. In this case you will get an error message.

A bulk deletion of domain contacts is possible.

Examples for XML Requests and Answers

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0303</code>
    <handle>
      <id>9926612</id>
    </handle>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0303</code>
      <text>Deletion of the domain contact has been started.</text>
      <type>notify</type>
    </status>
  </result>
</response>
```

13.6 Contact Inquire Tasks (0304)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
address		yes		yes		yes	yes
city	City	yes		yes			yes
country		yes		yes		yes	yes

Value	Description	Info	Info <key>	List	List <key>	order	where
created	Creation date	yes		yes		yes	yes
domainsafe	Is this contact secured by DomainSafe? Values: true/false	yes					
email	Email address	yes			yes	yes	yes
fax	Fax number	yes		yes		yes	yes
fname	Firstname	yes		yes		yes	yes
id	Contact ID	yes		yes		yes	yes
lname	Lastname	yes		yes		yes	yes
organization	Organization	yes		yes		yes	yes
owner	Object owner	yes		yes		yes	yes
pcode	Postal code	yes		yes		yes	yes
phone	Phone number	yes		yes		yes	yes
protection	Data, to be displayed in a Whois request	yes					
state	State	yes		yes		yes	yes
title	Title	yes		yes		yes	yes
type	Type (PERSON, ORG, ROLE)	yes		yes		yes	yes
updated	Last updated	yes		yes		yes	yes
verification	The status of the domain contact verification.	yes		yes		yes	yes
sip	SIP address	yes			yes	yes	yes
updater	User who made the last update	yes			yes	yes	yes

13.7 Contact Info (0304)

With this task you can inquire the domain contact data of one or several contacts. With the single inquire you receive the details of a single domain contact.

Examples for XML Requests and Answers

NOTE

You can enter the id or the alias key. If the value of the alias and/or id key is not found, a list inquire will be started.

Example Request(single inquire)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0304</code>
    <handle>
      <id>9925345</id>
      <alias>Maria Doe</alias>
    </handle>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <handle>
        <alias></alias>
        <type>PERSON</type>
        <fname>Peter</fname>
        <lname>Doe</lname>
        <title></title>
        <organization>PDA</organization>
        <address>Example Street 3</address>
        <pcode>12345</pcode>
        <city>Example City</city>
        <state></state>
        <country>USA</country>
        <phone>+49+1-12345-12345</phone>
        <fax>+49+1-12345-12345</fax>
        <email>Doe@example.com</email>
        <sip></sip>
        <protection>B</protection>
        <nic_ref>de</nic_ref>
        <nic_ref>com</nic_ref>
        <nic_ref>net</nic_ref>
        <remarks></remarks>
        <extension>
          <gender></gender>
          <idnumber></idnumber>
          <vatnumber></vatnumber>
          <trademark_name></trademark_name>
          <trademark_number></trademark_number>
          <trademark_country></trademark_country>
          <trademark_regdate></trademark_regdate>
          <trademark_appdate></trademark_appdate>
          <companynumber></companynumber>
          <birthday></birthday>
          <birthplace></birthplace>
          <birthcountry></birthcountry>
        </extension>
      </data>
      <status>
        <code>S0304</code>
        <text>Domain contact information was inquired successfully.lt.</text>
        <text></text>
        <type>success</type>
      </status>
    </result>
  </response>
```

13.8 Contact List (0304)

With this task you can you receive a list of all your contacts with certain details.

Examples for XML Requests and Answers

Example Request(list inquire)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0304</code>
    <view>
      <offset>0</offset>
      <limit>10</limit>
      <children>0</children>
    </view>
    <where>
      <key>name</key>
      <operator>like</operator>
      <value>Example*</value>
    </where>
    <key>created</key>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <handle>
        <alias>Example Company</alias>
        <created>2008-11-21 09:45:46</created>
        <fname>Peter</fname>
        <id>9910873</id>
        <lname>Doe</lname>
        <organization>Example Company 1</organization>
        <owner>
          <context></context>
          <user>customer</user>
        </owner>
        <type>ORG</type>
      </handle>
      <handle>
        <alias>Maria Doe</alias>
        <created>2008-11-17 15:15:13</created>
        <fname>Maria</fname>
        <id>9910825</id>
        <lname>Muster</lname>
        <organization>Example Company 2</organization>
        <owner>
          <context></context>
          <user>customer</user>
        </owner>
        <type>PERSON</type>
      </handle>
    </data>
    <summary>2</summary>
```

```

</data>
<status>
  <code>S0304</code>
  <text></text>
  <text>Domain contact information was inquired successfully.</text>
  <type>success</type>
</status>
</result>
</response>

```

13.9 Additional Information for Domain Contacts (Extensions)

Domain contact extensions are additional keys for the domain contacts which are required when registering certain TLDs.

Key	Occurs	Format	Description
handle / extension / mode	0 - 1	enum	This key determines how the extension data will be handled by the update. Mode is only possible for <i>ContactUpdate</i> . <ul style="list-style-type: none"> replace: existing domain contact data will be replaced merge: existing domain contact data will be added Default = merge.
handle / extension / gender	0 - 1*)	enum	Gender <ul style="list-style-type: none"> "" = default m f
handle / extension / travel_uin	0 - 1 *)	string	"Travel Unique Identifying Number" (.travel only)
handle / extension / idnumber	0 - 1*)	string	Identification number of the personal id card
handle / extension / vatnumber	0 - 1*)	string	VAT number
handle / extension / trademark_name	0 - 1*)	string	Trademark name
handle / extension / trademark_number	0 - 1*)	string	Trademark number
handle / extension / trademark_country	0 - 1*)	country	Country, where the trademark is valid
handle / extension / trademark_regdate	0 - 1*)	date	Registration date of the trademark
handle / extension / trademark_appdate	0 - 1*)	date	Application date of the trademark (yyyy-MM-dd)
handle / extension / companynumber	0 - 1*)	string	Company number
handle / extension / birthday	0 - 1*)	date	Date of birth (yyyy-MM-dd)
contact/extension / birthcode	0 -1 *)	string	Postal code of the birth place
handle / extension / birthplace	0 - 1*)	string	Birth place

Key	Occurs	Format	Description
handle / extension / birthcountry	0 - 1	country	Country of birth (ISO 3166 Country Code)
handle / extension / language	0 - 1	string	Contact language
handle / extension / mobile_phone	0 - 1	string	Mobile phone number

*) Depends on the registry.

13.10 Extensions for .aero Domains

An .aero domain can be registered with an ENS Auth ID and an ENS Key (Link to apply for an aero-ID

http://www.information.aero/registration/manage_your_aero_id/apply

Key	Occurs	Format	Description
handle / extension / aero_ens_auth_id	1	string	Authentication ID
handle / extension / aero_ens_key	1	string	Aero ENS key

13.11 Extensions for .au domains (Australia)

Key	Occurs	Format	Description
handle / extension / au_registrant_id	1	int	"Australian Business Number" or "Registered Business Number".
handle / extension / au_registrant_id_type		enum	Concerns the Registrant ID which you have selected in the Registrant ID form field. Select the matching entry in the list. <u>Available values:</u> <ul style="list-style-type: none"> ▪ ACN ▪ ABN ▪ OTHERS
handle / extension / au_eligibility_type	1	enum	The type of entity that the registrant is. <u>Available values:</u> <ul style="list-style-type: none"> ▪ COMPANY ▪ REGISTERED_BUSINESS ▪ SOLE_TRADER ▪ PARTNERSHIP ▪ TRADEMARK_OWNER ▪ PENDING_TM_OWNER ▪ CITIZEN_RESIDENT

Key	Occurs	Format	Description
			<ul style="list-style-type: none"> ▪ INCORPORATED_ASSOCIATION ▪ UNINCORPORATED_ASSOCIATION ▪ CLUB ▪ NON_PROFIT_ORGANISATION ▪ CHARITY ▪ TRADE_UNION ▪ INDUSTRY_BODY ▪ COMMERCIAL_STATUTORY_BODY ▪ POLITICAL_PARTY ▪ RELIGIOUS_CHURCH_GROUPS
handle / extension / au_eligibility_name	1	string	This field is usually optional and whether it is required is determined by your policy reason for the domain registration, if in doubt just use the same details as your Registrant Name.
handle / extension / au_eligibility_ID	1	int	The corresponding ID number for the eligibility name if applicable, if in doubt just use the same details as your Registrant ID.
handle / extension / au_eligibility_ID_type	1	enum	<p>The type of Eligibility ID as above.</p> <p>Available values:</p> <ul style="list-style-type: none"> ▪ NSW_BN ▪ SA_BN ▪ NT_BN ▪ WA_BN ▪ TAS_BN ▪ ACT_BN ▪ QLD_BN ▪ TM ▪ OTHER
handle / extension / au_policy_reason	1	enum	<p>Reason for eligibility.</p> <p>1: All SLD Domain Name is an exact match, abbreviation or acronym of the registrants name or trade mark.</p> <p>2: com.au, net.au For commercial purposes. Includes commercial entities, currently registered and trading in Australia, as well as commercial products and services. Due to technical reasons, please insert your Australian trademark name and number into the respective form fields of the form section ' Trademark' above.</p>
au_registrant_name	0 - 1	string	The registrants name

13.12 Extensions for .barcelona domains

Key	Occurs	Format	Description
handle / extension / /barcelona_intended_use	0 - 1	string	The statement of intended use for the domain name.

13.13 Extensions for .cat Domains (Catalan language area)

A .cat domain is registered by a deferred check up. As requestor the registry will send you the registration policies via email. If you confirm the policies the domain will be assigned to you. Later the registry will check if the use of the domain is according to the intended use.

Key	Occurs	Format	Description
handle / extension / cat_intended_use	1	string	Mandatory. Description of the intended use of the domain. This field is required for both ways of registration. See http://domini.cat/en_com_registrar.html#conditions .

13.14 Extensions for .hk Domains

Key	Occurs	Format	Description
contact / extension / hk_document_type	1	string	For persons the following document types exist: <ul style="list-style-type: none"> ▪ HKID: Hong Kong Identity Card ▪ OTHID: Others Country / Region Identity Card ▪ PASSNO: Passport ▪ IRTHCERT: Birth Certificate OTHIDV: Others Individual Document For organizations the following document types exist: <ul style="list-style-type: none"> ▪ CI: Certificate of Incorporation CRS: Certificate of Registration of a School ▪ HKSARG: Hong Kong Special Administrative Region Government Department ▪ HKORDINANCE: Ordinance of Hong Kong ▪ OTHORG: Other Organizations
contact / extension / hk_others	0 - 1	string	Additional descriptions for OTHIDV and OTHORG
contact / extension / hk_document_number	0 - 1	string	Document number
contact / extension / hk_document_origin	0 - 1	string	Country of licensure
contact / extension /	0 - 1	enum	For "Person" only.

Key	Occurs	Format	Description
above_18			Is the person 18 years of age or older. Values: <ul style="list-style-type: none"> ▪ yes: the person is 18 years of age or older ▪ no: the person is younger than 18 You have to select an option.
contact / extension / industry_type	0 - 1	int	Industry types A list with the numbers and respective industry types is in the addendum

13.15 Extensions for .it Domains

Key	Occurs	Format	Description
contact / extension / it_entity_type	1	string	Enter the number of the matching entity: <i>1= Italian and foreign natural persons</i> <i>7= Foreign non natural persons</i> <i>2= Italian companies/one man companies</i> <i>3= Italian freelance workers/professionals</i> <i>4= Italian non-profit organizations</i> <i>5= Italian public organizations</i> <i>6= Italian other subjects</i>

13.16 Extensions für .jobs Domains

Key	Occurs	Format	Description
contact / extension / jobs_contact_title	0 - 1	string	Title of the domain contact in the organization.
contact / extension / jobs_website	1	string	Website
contact / extension / jobs_industry_class	0 - 1		Industry class. Enter the value of the industry class. Industry classes list: 2: Accounting/Banking/Finance 3: Agriculture/Farming 21: Biotechnology/Science 5: Computer/Information Technology 4: Construction/Building Services

Key	Occurs	Format	Description
			12: Consulting 6: Education/Training/Library 7: Entertainment 13: Environmental 19: Hospitality 10: Government/Civil Service 11: Healthcare 15: HR/Recruiting 16: Insurance 17: Legal 18: Manufacturing 20: Media/Advertising 9: Parks & Recreation 26: Pharmaceutical 22: Real Estate 14: Restaurant/Food Service 23: Retail 8: Telemarketing 24: Transportation 25: Other
contact / extension / jobs_hr_member	0 - 1		Choose whether the contact is is a member of the „Human Resource Organization“. <ul style="list-style-type: none"> • Yes = 1 • No = 0
contact / extension / jobs_admin_type	1		Here you enter whether the contact is the administrative contact with the permission to act as an independent agent /recruiter on behalf of the listed registrant. <ul style="list-style-type: none"> • Yes = 1 • No = 0

13.17 Extensions for .ro Domains

Key	Occurs	Format	Description
contact / extension /ro_person_type	1	string	Enter the character(s)of the matching entity: P = Private Person AP = Autorisierte Person N =C Nicht-kommerzielle Organisation C = Kommerziell GI = Regierungseinrichtung PI = Öffentliche Einrichtung O = Andere juristische Personen

13.18 Extensions for .ru Domains (Russia)

For the registration of .ru-domains detailed information about the ID card of the domain owner is required.

Key	Occurs	Format	Description
contact / extension / id_authority	1	string	ID authority
contact / extension / id_date_of_issue	1	string	Date of issue. Format: yyyy-MM-dd
handle / extension / id_valid_till	1	string	Validation data. Format: yyyy-MM-dd

13.19 Extensions for .swiss domains

Key	Occurs	Format	Description
handle / extension /swiss_enterprise_id	0 - 1	string	The enterprise ID, in the specific context of .swiss based on current rules, is the Swiss UID/IDE/IDI. The full name is in German "Unternehmens-ID", in French "Numéro d'identification des entreprises (IDE)" and in Italian "Numero d'identificazione delle imprese" (IDI).
handle / extension /swiss_intended_use	0 - 1	string	The statement of intended use for the domain name.

13.20 Extensions for .uk Domains

The <uk_type> extension was introduced to categorize the organization types. If you want certain contact data not to be displayed in the Whois, use the organization types IND or FIND.

Key	Occurs	Format	Description
contact / extension / uk_type	0 – 1*)	string	For the sub domains ltd.uk, net.uk and plc.uk the selection of <uk_type> is mandatory. For co.uk and me.uk the selection of <uk_type> is not mandatory. List of organization types: <ul style="list-style-type: none"> • LTD - UK Limited Company • PLC - UK Public Limited Company • PTNR - UK Partnership • STRA - UK Sole Trader • LLP - UK Limited Liability Partnership • IP - UK Industrial/Provident Registered Company • IND - UK Individual (representing self)

Key	Occurs	Format	Description
			<ul style="list-style-type: none"> • SCH - UK School* RCHAR - UK Registered Charity • GOV - UK Government Body • CRC - UK Corporation by Royal Charter • STAT - UK Statutory Body • OTHER - UK Entity that does not fit into any of the above (e.g. clubs, associations, many universities) • FIND - Non-UK Individual (representing self) • FCORP - Non-UK Corporation • FOTHER - Non-UK Entity that does not fit into any of the above (e.g. charities, schools, clubs, associations)

*) depends on the subdomain

13.21 Extensions for .xxx Domains

Key	Occurs	Format	Description
contact / extension / membership_id	0 – 1	string	Membership ID (token) for the .XXX community. You receive your ID after you have finished the ICM registry membership process successfully and having created an account there.
contact / extension / non_resolver_domain	0 - 1	string	<p>XXX community membership. Members have a membership ID (token).</p> <p><u>Available values:</u> true = member false = no member</p>

14 Domain Contact Verification

The WHOIS Accuracy Specification requires the active verification of each domain contact used as the domain owner contact (OwnerC) for gTLDs. This verification is carried out per email.

After registering a gTLD, the domain owner (OwnerC) receives an email with a verification request. This email contains a link leading to a website where the email address must be actively confirmed. Each domain contact used as the domain owner contact (OwnerC) only needs to be verified once. It can then be used without further restrictions for all all gTLDs. If the verification is not carried out within the specified period, the domain status is changed to HOLD at the registry. This means that the domain cannot be resolved by the name servers and will be unavailable.

Deadlines for deactivation in the case of unconfirmed OwnerC domain contacts

- after DOMAIN CREATE: 15 days
- after DOMAIN TRANSFER: 30 days
- after DOMAIN OWNERCHANGE: 30 days
- after UPDATE of the domain contact email address: 30 days

14.1 Description of the XML Keywords

XML Key	Occurs	Format	Description
code	1	task	Contact Verification Create 0341 Contact Verification Info 0344 Contact Verification with Alias Info 0344 Contact Verification List 0345 Contact Verification Resend Email 0349
handle_verification	1		
handle_verification / handle	1	string	
handle_verification /handle / id	1	string	Domain contact Id of the contact that is to be verified.

14.2 Contact Verification Create (0341)

With this task you start a request for the domain contact verification.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0341</code>
    <handle_verification>
      <handle>
        <id>112233</id>
      </handle>
    </handle_verification>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data />
    <status>
      <code>S0341</code>
      <type>success</type>
      <object>
        <type>handle_verification</type>
        <value>112233</value>
      </object>
    </status>
  </result>
</response>
```

14.3 Contact Verification Inquire Tasks (0344, 0345)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
confirmed	Date of the verification confirmation	yes		yes		yes	yes
created	Creation date of the verification request	yes		yes		yes	yes
name	The domain for which the contact is used as OwnerC	yes		yes			
failed	Date of the verification refusal	yes		yes		yes	yes
handle_id	Domain contact id of the contact that is to be verified.	yes		yes			
message_sent	Sending date of the message	yes		yes		yes	yes
updated	Last updated	yes		yes		yes	yes
confirm_ip	The IP of the system on which the confirmation link in the confirmation email was clicked. Only for confirmed verification.	yes					
handle	Contact data	yes					
comment	Comment (only when refused)	yes					
action	Verification trigger	yes					
owner	Domain owner	yes					
updater	User who made the last update	yes					
mail	System email for the user		yes				

14.4 Contact Verification Info (0344)

With this task you can inquire the data of a verification request.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0344</code>
    <handle_verification>
      <handle>
        <id>11404405</id>
      </handle>
    </handle_verification>
  </task>
</request>
```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <handle_verification>
        <handle>
          <alias></alias>
          <type>PERSON</type>
          <fname>Peter</fname>
          <lname>MusterDoe</lname>
          <title></title>
          <organization>PDA</organization>
          <address>MusterstrasseExample Street 3</address>
          <pcode>12345</pcode>
          <city>MusterstadtExample City</city>
          <state></state>
          <country>DeutschlandUSA</country>
          <phone>+49+1-12345-12345</phone>
          <fax>+49+1-12345-12345</fax>
          <email>MusterDoe@example.com</email>
          <protection>B</protection>
          <extension />
          <verification>SUCCESS</verification>
          <owner>
            <user>customer</user>
            <context></context>
          </owner>
          <updater>
            <user>customerlroot</user>
            <context></context>
          </updater>
          <id>11404405</id>
          <created>2013-07-10 12:33:32</created>
          <updated>2013-07-10 12:33:31</updated>
        </handle>
        <confirmed>2013-12-12 12:57:05</confirmed>
        <confirm_ip>125.11.0.12</confirm_ip>
        <message_send>2013-12-12 11:52:01</message_send>
        <domain>
          <name>example.com</name>
          <status>SUCCESS</status>
          <deactivation>2013-12-12 11:44:03</deactivation>
        </domain>
        <message>
          <message_id>
            <id>2054</id>
          </message_id>
          <status>SUCCESS</status>
        </message>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <updater>
          <user>customer</user>
          <context>4</context>
        </updater>
        <created>2013-12-12 11:39:11</created>
        <updated>2013-12-12 11:52:21</updated>
      </handle_verification>
    </data>
    <status>
      <code>S0344</code>
      <text>S0344</text>
      <type>success</type>
    </status>
  </result>
</response>

```

```

        <type>handle_verification</type>
        <value>21397922</value>
      </object>
    </status>
  </result>
  <stid>20131212-app1-dev-4062</stid>
</response>

```

14.5 Contact Verification List (0345)

With this task you can inquire the data of the verification requests.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0345</code>
    <view>
      <offset>0</offset>
      <limit>10</limit>
      <children>0</children>
    </view>
    <where>
      <key>handle.id</key>
      <operator>like</operator>
      <value>1142*</value>
    </where>
    <key>created</key>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <summary>2</summary>
      <handle_verification>
        <handle>
          <alias>customer1_alias</alias>
          <id>11404405</id>
        </handle>
        <domain>
          <name>example</name>
          <status>PENDING</status>
        </domain>
        <created>2012-11-08 13:19:36</created>
        <updated>2013-11-08 15:36:58</updated>
      </handle_verification>
      <handle_verification>
        <handle>
          <alias>customer2_alias</alias>
          <id>11404405</id>
        </handle>

```



```

        <domain>
            <name>example1</name>
            <status>PENDING</status>
        </domain>
        <created>2012-11-08 13:19:36</created>
        <updated>2013-11-08 15:36:58</updated>
    </handle_verification>
</data>
<status>
    <code>S0345</code>
    <type>success</type>
</status>
</result>
</response>

```

14.6 Contact Verification Resend Email (0349)

With this task you can resend the verification request.

Examples for XML Requests and Answers

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0349</code>
    <handle_verification>
      <handle>
        <id>11404406</id>
      </handle>
    </handle_verification>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8"?>
<response>
  <result>
    <data/>
    <status>
      <code>S0349</code>
      <type>success</type>
      <object>
        <type>handle_verification</type>
        <value>11404406</value>
      </object>
    </status>
  </result>
</response>

```

15 Redirects

15.1 Description of the XML Keywords

Schlüsselwort	Occurs	Format	Description
code	1	task	Redirect Create 0501 Redirect Update 0502 Redirect Delete 0503 Redirect Inquire 0504
redirect			
redirect / type	1	enum	Redirect type <ul style="list-style-type: none"> • domain • email
redirect / mode	1	enum	Redirect mode Domain redirect: <ul style="list-style-type: none"> • frame = the URL of the source domain will be displayed in the browsers address bar • http = the URL of the target domain will be displayed in the browsers address bar (HTTP 301 Redirect). • https =the URL of the target domain will be displayed in the browsers address bar. The https protocol is used (HTTP 301 Redirect). Email redirect: <ul style="list-style-type: none"> • single = emails of the specified email account will be redirected • catchall = emails of all email accounts of the source domain will be redirected. For "catchall" enter the address without local part, e.g. "@example.com".
redirect / source	1	string	The domain(s) to be redirected. Enter the domain with or without "www".
redirect / target	1	string	The URL of the target domain. Enter the domain without "http://"
redirect / title	0 - 1	string	For frame mode only. The title to be shown in the title bar of the browser.
redirect / backup	0 - 3	string	Enter the backup targets here. If the first target is not accessible, the domain or email address will be redirected to the backup target automatically. For frame mode only.

15.2 Redirect Create (0501)

With this task you can create a new domain or email redirect.

IMPORTANT NOTE

Redirects are managed by a dedicated server (IP address 62.116.130.8). In addition to the entry in the redirector, you have to change the domain and the subdomain entries to point to this IP address. If you run your own nameservers, you have to enter the IP address manually on your nameserver. If you want to setup an email redirect without web redirect, you have to change the MX of the source. It has to refer to the IP Address 62.116.130.8. Please change the IP address of the subdomain which is used as MX.

Examples

Redirecting a domain:

- *example.com => example.de*
- *example.com => example.de/subfolder/index.html*

Redirecting email addresses

- *jane_smith@example.com => jane@example.de (mode=single)*
- *@example.com=> jane@example.de (mode=catchall)*

Domains are treated case-sensitive. Email addresses are automatically converted to lower case by the system.

Examples for XML Requests and Answers

Example Request (Http)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0501</code>
    <redirect>
      <type>domain</type>
      <mode>http</mode>
      <source>www.source-example.com</source>
      <target>www.target-example.com</target>
    </redirect>
  </task>
</request>
```

>

Example Response (Http)

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0501</code>
      <text>Redirect has been created successfully.</text>
```

```

        <type>success</type>
    </status>
</result>
</response>

```

Example Request (Frame)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0501</code>
    <redirect>
      <type>domain</type>
      <mode>frame</mode>
      <title>Redirection-Title</title>
      <source>www.source-example.com</source>
      <target>www.target-example.com</target>
      <backup>www.backup-target1.de</backup>
      <backup>www.backup-target2.de</backup>
      <backup>www.backup-target3.de</backup>
    </redirect>
  </task>
</request>

```

Example Response(Frame)

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0501</code>
      <text>Redirect has been created successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

Example RequestEmail (single)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0501</code>
    <redirect>
      <type>email</type>
      <mode>single</mode>
      <source>source@email-address.com</source>
      <target>target@email-address.com</target>
    </redirect>
  </task>
</request>

```

Example Response Email (single)

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0501</code>
      <text>Redirect has been created successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

Example Request Email (Catchall)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0501</code>
    <redirect>
      <type>email</type>
      <mode>catchall</mode>
      <source>@email-address.com</source>
      <target>target@email-address.com</target>
    </redirect>
  </task>
</request>
```

Example ResponseE-Mail (Catchall)

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0501</code>
      <text>Redirect has been created successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

15.3 Redirect Update (0502)

With this task you can actualize an existing domain or email redirect. You can change the target (target), the mode (mode) or add backup-targets (backup).

NOTE

The structure of redirect updates and redirect creates are similar. Every value, except "source" and "type" can be changed; "type" must not be changed, because URLs normally are not valid email addresses and email addresses normally are not valid URLs.

Examples for XML Requests and Answers

Example Request (Domain)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0502</code>
    <redirect>
      <type>domain</type>
      <mode>frame</mode>
      <title>New title</title>
      <source>www.source-example.com</source>
      <target>www.target-domain.de/somewhere</target>
      <backup>www.target-example-backup.com/somewhere/else</backup>
    </redirect>
  </task>
</request>
```

Example Response(Domain)

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0502</code>
      <text>Redirect has been updated successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

Example Request (Email)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0502</code>
    <redirect>
      <type>email</type>
      <mode>single</mode>
      <source>source@email-address.de</source>
      <target>new_target@email-address.net</target>
    </redirect>
  </task>
</request>
```

Example Response(Email)

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0502</code>
      <text>Redirect has been updated successfully.</text>
    </status>
  </result>
</response>
```

```

        <type>success</type>
    </status>
</result>
</response>

```

15.4 Redirect Delete (0503)

With this task you can delete an existing domain or email redirect.

NOTE

The following examples have a type tag which is optional. If there is no such tag, the type will be determined automatically. For domain redirects pay attention to the upper and lower cases. Email addresses however will be converted to lower cases.

Examples for XML Requests and Answers

Example Request(Domain)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0503</code>
    <redirect>
      <type>domain</type>
      <source>www.source-domain.de</source>
    </redirect>
  </task>
</request>

```

Example Response(Domain)

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0503</code>
      <text>Redirect has been deleted successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

Example Request(Email)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0503</code>
    <redirect>
      <type>email</type>

```

```

    <source>source@email-address.com</source>
  </redirect>
</task>
</request>

```

Example Response(Email)

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S0502</code>
      <text>Redirect has been deleted successfully. </text>
      <type>success</type>
    </status>
  </result>
</response>

```

15.5 Redirect Inquire Tasks (0504)

This task allows you to inquire the data of your current redirects. With the single inquire you receive the details of a single domain contact.

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
backup	Backup targets for the redirect	yes					
created	Creation date of the redirect	yes		yes		yes	yes
mode	Redirect type	yes		yes		yes	yes
owner	Owner of the redirected domain	yes			yes	yes	yes
source	Redirected domain	yes		yes		yes	yes
target	Target email address of the redirect	yes		yes		yes	yes

Value	Description	Info	Info <key>	List	List <key>	order	where
title	Frame title only for frame redirects.	yes				yes	yes
type	Domain or email redirect.	yes		yes		yes	yes
updated	Last updated.	yes		yes		yes	yes
updater	User who made the last redirect.	yes			yes	yes	yes

15.6 Redirect Info (0504)

This task allows you to inquire the data of a current redirect.

Examples for XML Requests and Answers

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0504</code>
    <redirect>
      <source>www.source-domain.de</source>
    </redirect>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <redirect>
        <owner>
          <user>customer</user>
          <context></context>
        </owner>
        <type>domain</type>
        <mode>frame</mode>
        <title>Titel</title>
        <source>www.source-domain.de</source>
        <target>www.target-example.com</target>
        <backup>www1.target-example.com</backup>
        <backup>www2.target-example.com</backup>
        <backup>www3.target-example.com</backup>
        <updated>2009-05-22 10:40:55</updated>
        <created>yyyy-MM-dd hh:mm:ss</created>
        <created>2009-05-22 10:40:55</created>
        <updated>2009-05-22 10:40:55</updated>
      </redirect>
    </data>
    <status>
      <text>Redirect data was inquired successfully.</text>
```

```

        <type>success</type>
        <code>S0504</code>
    </status>
    </result>
</response>

```

15.7 Redirect List (0504)

This task allows you to inquire the data of your current redirects.

Examples for XML Requests and Answers

Example Request(list inquire)

```

<?xml version="1.0" encoding="utf-8" ?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0504</code>
    <view>
      <offset>0</offset>
      <limit>30</limit>
      <children></children>
    </view>
    <key>updated</key>
    <key>created</key>
    <key>title</key>
    <where>
      <and>
        <key>type</key>
        <operator>eq</operator>
        <value>domain</value>
      </and>
      <and>
        <key>source</key>
        <operator>like</operator>
        <value>e< *value>
      </and>
    </where>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <summary>2</summary>
      <redirect>
        <owner>
          <user>customer2</user>
          <context></context>
        </owner>
        <updater>
          <user>customer</user>
          <context>1018945</context>
        </updater>
        <type>domain</type>

```

```
<mode>http</mode>
<source>example.com</source>
<target>example2.com</target>
<updated>yyyy-MM-dd hh:mm:ss</updated>
<created>yyyy-MM-dd hh:mm:ss</created>
</redirect>
<redirect>
  <owner>
    <user>customer3</user>
    <context></context>
  </owner>
  <updater>
    <user>customer</user>
    <context>1018945</context>
  </updater>
  <type>domain</type>
  <mode>frame</mode>
  <source>example3.com</source>
  <target>example4.com</target>
  <updated>yyyy-MM-dd hh:mm:ss</updated>
  <created>yyyy-MM-dd hh:mm:ss</created>
</redirect>
</data>
<status>
  <text>Redirect data was inquired successfully.</text>
  <type>success</type>
  <code>S0504</code>
</status>
</result>
</response>
```

16 User

16.1 Description of the XML Keywords

Key	Occurs	Format	Description
code	1	task	User Create 1301001 User Update 1301002 User Inquire 1301004 User Profile Update 1301014 User Copy 1301016 Object User_Assignment 1308 Object User_Assignment_All 1308001
user	1		
user / user	1	string	The user name
User / context	1	string	ID of the user. Default:
user / ip_restrictions	0 - 1		IP net of the user. Stating the IP-network is intended to increase the security and prevent access to the system from other networks (for example: foreign countries).
user / ip_restrictions / ip_restriction	0 - n		Grouping object
user / ip_restrictions / ip_restriction / address	1	string	Subnet with mit CIDR , e.g. 192.168.0.254/32
user / ip_restrictions / ip_restriction / type	1	enum	CONTEXT, USER, CLONE
user / ip_restrictions / ip_restriction / updated	1	date	Last updated
user / password	1	string	The user password of the new user.
user / status	1	enum	The status of the new user <ul style="list-style-type: none"> • 0= master • 1= admin • 2= normal • See the user status description in the addendum.
user / substatus	1	enum	The sub-status of the user. <ul style="list-style-type: none"> • 0= + • 1= * • 2= "" • 3= clone • See the user status description in the addendum.
user / default_mail	1	string	Default email address of the new user.
user / language	1	enum	The language for the new user. It will be used for the user interface and the system messages. <ul style="list-style-type: none"> • en • de • es

Key	Occurs	Format	Description
user / owner / user	1	string	The user name of the parent
user / owner / context	1	string	The ID of the parent.
code	1	task	User Profile Update 301014
profile	1		User profile
profile /name	1	string	Name of the profile group
profile /item	1 - n		Keyword
profile /item / key	1	string	Name of the key
			NOTE Keys, which are not defined in the request, will be deleted.
profile /item / value	1	string	Key value
profile /item / flag	1	string	Here you define how the profile data will be assigned to the subusers. <ul style="list-style-type: none"> • optional: the profile data will not be assigned to the subusers. • recurse: the defaults will be submitted by the subuser. They are editable. • fix: the defaults will be submitted by the subuser. They are not editable.
object	1 - n	string	Object (Object user assignment)
object / type	1	enum	Object type <u>Available values</u> <ul style="list-style-type: none"> • domain • handle • zone • domainzone (domain and zone)
object / name	1	string	Name of the object which will be assigned
object / owner	1	string	Target user
object / owner / user	1	string	Name of the target user
object / owner / context	1	string	ID of the personalized system of the target user
cancelation	0 - 1	enum	Treatment of existing cancelations. For object user assignment tasks. <u>Available values:</u> <ul style="list-style-type: none"> ▪ assign= existing cancelation request will be assigned to the new user. ▪ delete = existing cancelation requests will be deleted If you do not use this key, "assign" will be used as default.
task / acl	1		

Key	Occurs	Format	Description
task / acl / code	1	integer	Task code, the ACL. refers to
task / acl / user	1	string	The user who assigns the ACLs.
task / acl / entry	1	string	
task / acl / entry / locked		enum	1 = lock 0 = unlock
task / acl / entry / children		enum	1 = lock subusers too 0 = do not lock subusers
task / acl / entry / child_add	0 - n		Subuser to whom the ACL is granted
task / acl / entry / child_add /user	1	string	Name of the subuser the ACL is granted to.
task / acl / entry / child_add /context	1	integer	Context of the subuser the ACL is granted to.
task / acl / entry / child_rem	0 - n		Subuser from which the ACL is removed from
task / acl / entry / child_rem / user	1	string	Name of the subuser the right is removed from.
task / acl / entry / child_rem /context	1	integer	Context of the subuser the right is removed from.
task / acl / user	1	string	The user who grants the ACL.
task / acl / user / name	1	string	Name of the user who grants the ACL.
task / acl / user / context	1	integer	Context of the user who grants the ACL.
task / clone			Only for the user copy task 1301016
task / clone /source	1		User who will be copied.
task / clone /source /user	1	string	Name of the user who will be copied
task / clone /source / context	1	integer	Context of the user who will be copied.
task / clone /target	1		Target user that is generated by the copy function.
task / clone /target /user	1	string	Name of the target user
task / clone /target /default_mail	1	string	Default email address of the target user

16.2 User Create (1301001)

With this task you can create a new user.

Examples for XML Requests and Answers

Main user creates a direct subuser

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>4</context>
  </auth>
  <task>
    <code>1301001</code>
    <user>
      <user>sub-customer</user>
      <password>password</password>
      <status>2</status>
      <substatus>2</substatus>
      <default_mail>sub-customer@example.com</default_mail>
      <language>de</language>
    </user>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S1301001</code>
      <object>
        <type>user</type>
        <value>sub-customer, 4</value>
      </object>
      <text>User was successfully created.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

Main user creates a direct subuser. Request with an automatic password generation feature.

NOTE

If you do not enter a password when creating a new user, a password with a length of 15 characters will be automatically generated by the system. The password is then displayed in the response message or in a visual cue in the top right-hand corner when using the graphical interface.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>4</context>
  </auth>
  <task>
    <code>1301001</code>
    <user>
      <user>sub-customer</user>
      <status>2</status>
      <substatus>2</substatus>
      <default_mail>sub-customer@example.com</default_mail>
      <language>de</language>
    </user>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data/>
    <msg>
      <text>The password was automatically generated by the system.</text>
      <code>MSG13004</code>
      <type>notice</type>
      <object>
        <type>password</type>
        <value>/WKU;3Pm3%</value>
      </object>
    </msg>
    <status>
      <code>S1301001</code>
      <object>
        <type>user</type>
        <value>sub-customer, 4</value>
      </object>
      <text>User was successfully created.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

Main user creates a new user for his subuser

Example Request

```
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>4</context>
  </auth>
  <task>
    <code>1301001</code>
    <user>
      <user>NewSubUser</user>
      <password>password</password>
      <context>4</context>
      <status>2</status>
      <substatus>1</substatus>
      <default_mail>newuser@example.com</default_mail>
    </user>
  </task>
</request>
```



```

        <language>de</language>
        <owner>
            <user>user</user>
            <context>4</context>
        </owner>
    </user>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" >
<response>
  <result>
    <status>
      <code>S1301001</code>
      <object>
        <type>user</type>
        <value>NewSubUser, 4</value>
      </object>
      <text>User was successfully created.</text>
      <type>success</type>
    </status>
  </result>
  <stid>20140318-appl-test-12644</stid>
</response>

```

16.3 User Update (1301002)

With this task you can edit the data of a user.

Example Request

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>1301002</code>
    <user>
      <user>customersub1</user>
      <context>104</context>
      <password>password</password>
      <status>2</status>
      <substatus>2</substatus>
      <default_mail>customersub1@example.de</default_mail>
      <language>de</language>
    </user>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S1301002</code>
      <text>User was successfully updated.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

16.4 User Inquire Tasks (1301004)

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
ancestor	All superior users	yes					
auth_type							
condition	Availability of the user Values: ok, expired, locked	yes		yes			
context	The Personal System ID	yes		yes		yes	yes
default_mail	Default email address of the user	yes		yes		yes	yes
language	Communication language	yes					
locked	Is the user locked? Values: true, false	yes		yes			
parent	The superior user	yes					
uid	Internal user ID	yes					
status	User status	yes		yes		yes	yes
substatus	Substatus of the user	yes		yes		yes	yes
user	User name	yes		yes		yes	yes
support_mail	Support email address of the reseller. Only for personalized system	yes					

Diese Werte werden in einem eigenen Block außerhalb des User-Blocks ausgegeben.

Value	Description	Info	Info <key>	List	List <key>	order	where
Acl	User privileges	yes					
Customer	Customer	yes					
Owner	Besitzer des Objektes	yes					
Profile	User profile data	yes					

16.5 User Info (0110004)

With this task you can inquire the data of the preregistration request including the status.

Example Request(single inquire)

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>1301004</code>
    <user>
      <user>customer2</user>
      <context>104</context>
    </user>
  </task>
</request>
```

Example Response

```
?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <user>
        <acl>
          <allowed>1</allowed>
          <children>1</children>
          <task>1301013</task>
        </acl>
        <acl></acl>
        <ancestors>104, 1, 0</ancestors>
        <autodelete_user>0</autodelete_user>
        <context>104</context>
        <default_mail>musterdoe@example.com</default_mail>
        <ip_restrictions>
          <address>192.168.0.253/32</address>
          <type>CLONE</type>
          <updated>2016-02-04 09:36:43</updated>
        </ip_restriction>
        <ip_restriction>
          <address>192.168.0.254/32</address>
          <type>CONTEXT</type>
          <updated>2016-02-04 09:36:43</updated>
        </ip_restriction>
      </user>
    </data>
  </result>
</response>
```

```

        <language>en</language>
        <owner>
          <context>104</context>
          <status>1</status>
          <substatus>0</substatus>
          <user>customer</user>
        </owner>
        <parent>104</parent>
        <profile>
          <item>
            <flag>OPTIONAL</flag>
            <key>account_holder</key>
            <value></value>
          </item>
          <name>bank</name>
        </profile>
        <profile></profile>
        <status>2</status>
        <substatus>0</substatus>
        <supervisor>0</supervisor>
        <uid>1019008</uid>
        <user>John</user>
      </user>
    </data>
    <status>
      <code>S1301004</code>
      <text>User data was inquired was successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

16.6 User List (1301004)

With this task you can inquire the data of several users.

Example Request(list inquire)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>1301004</code>
    <view>
      <limit>2</limit>
      <children>0</children>
      <offset>1</offset>
    </view>
    <key>uid</key>
    <key>user</key>
    <key>owner</key>
    <key>context</key>
    <key>default_mail</key>
    <key>status</key>
    <key>substatus</key>
    <key>language</key>
  </task>
</request>

```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <summary>10</summary>
      <user>
        <context>104</context>
        <default_mail>client2@example.es</default_mail>
        <language>de</language>
        <owner>104</owner>
        <status>2</status>
        <substatus>1</substatus>
        <uid>1019069</uid>
        <user>cliente2</user>
      </user>
      <user>
        <context>104</context>
        <default_mail>musterdoe@example.com</default_mail>
        <language>en</language>
        <owner>104</owner>
        <status>2</status>
        <substatus>0</substatus>
        <uid>1019008</uid>
        <user>John</user>
      </user>
    </data>
    <status>
      <code>S1301004</code>
      <text>User data was inquired successfully.</text>
      <text>.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

16.7 User Profile Update (1301014)

With this task you can actualize the user profile data.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context>104</context>
  </auth>
  <task>
    <code>1301014</code>
    <profile>
      <name>domain</name>
      <item>
        <key>nserver1</key>
        <value>ns1.example.net</value>
        <flag>OPTIONAL</flag>
      </item>
      <item>
        <key>nserver2</key>
        <value>ns2.example.net</value>
        <flag>OPTIONAL</flag>
      </item>
      <item>

```

```

        <key>ownerc</key>
        <value>123456</value>
        <flag>OPTIONAL</flag>
    </item>
    <item>
        <key>adminc</key>
        <value>123456</value>
        <flag>OPTIONAL</flag>
    </item>
    <item>
        <key>techc</key>
        <value>123456</value>
        <flag>OPTIONAL</flag>
    </item>
    <item>
        <key>zonec</key>
        <value>123457</value>
        <flag>OPTIONAL</flag>
    </item>
    <item>
        <key>zone_nsaction</key>
        <value>complete</value>
        <flag>OPTIONAL</flag>
    </item>
</profile>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S1301014</code>
      <text>Profile updated successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

16.8 User Copy (1301016)

With this task you can create a new user by copying a user.

Example Request

```

<?xml version="1.0"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>1301016</code>
    <clone>
      <source>
        <user>customer</user>
        <context></context>
      </source>
      <target>
        <user>targetuser1</user>
        <default_mail>customer@example.com</default_mail>
      </target>
    </clone>
  </task>
</request>

```

```

        </target>
    </clone>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<response>
  <result>
    <data/>
    <msg>
      <text>The password was automatically generated by the system.</text>
      <code>MSG13004</code>
      <type>notice</type>
      <object>
        <type>password</type>
        <value>}U6>ZSR*|e5mnf!</value>
      </object>
    </msg>
  </data>
  <status>
    <code>S1301016</code>
    <text>The user was copied successfully.</text>
    <type>success</type>
    <object>
      <type>user</type>
      <value>targetuser1</value>
    </object>
  </status>
</result>
<stid>20150723-appl-qm-487</stid>
</response>

```

16.9 User Delete (1301003)

With this task you can delete a user.

NOTE

A user can only be deleted if he has no objects (domains, zones, ...).

Example Request(single inquire)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>1301003</code>
    <user>
      <user>customersub1</user>
      <context></context>
    </user>
  </task>
</request>

```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S1301003</code>
      <text>User was successfully deleted.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

16.10 Object User Assignment (1308)

With this task you can assign a single object of one user to another one within your management.

Example Response

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth><task>
    <code>1308</code>
    <object>
      <type>handle</type>
      <name>239272</name>
      <!-- optional: target user -->
      <!--
      <owner>
        <user>newuser</user>
        <context></context>
      </owner>
      -->
    </object>
    <reply_to>EMAIL</reply_to>
    <cancelation>assign</cancelation>
  </task>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S1308</code>
      <text>The object user assignment was changed successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```


16.11 Object User Assignment All (1308001)

With this task you can assign all objects of one user to another user within your management.

Example Request

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>1308001</code>
    <source>
      <owner>
        <user>customer</user>
        <context></context>
      </owner>
    </source>
    <target>
      <owner>
        <user>target_user</user>
        <context></context>
      </owner>
    </target>
    <reply_to>customer@example.com</reply_to>
  </task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>S1308001</code>
      <text> </text>
      <type>success</type>
    </status>
  </result>
</response>
```

16.12 Object User Assignment Bulk (1308002)

With this task you can assign several objects of one user to another user within your management.

Example Response

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>1308002</code>
    <default>
      <type>domains</type>
    </default>
  </task>
</request>
```

```
        <owner>
            <user>customer</user>
            <context></context>
        </owner>
    </default>
    <object>
        <name>example.com</name>
    </object>
    <object>
        <name>example.org</name>
    </object>
    <object>
        <name>example.info</name>
    </object>
</task>
</request>
```

Example Response

```
<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <status>
      <code>1308002</code>
      <text>The object user assignments were changed successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>
```

17 Inquiring Requests

17.1 Description of XML Keywords

Key	Occurs	Format	Description
code	1	task	History Inquire 0713
view	1		The objects to be inquired
view / offset	0 - 1	string	The position in the list that the inquire should start listing the objects.
view / limit	0 - 1	string	Maximum number of objects to be inquired
view / children	0 - 1	enum	Inquire subusers: <ul style="list-style-type: none"> • 1 = subusers will be inquired • 0 = subusers will be not inquired
where	0 - 1	string	The <where> element contains the description of the object to be queried. In the <where> element you can interleave multiple search terms by using the keys <where>*/<and>”and <where>*/<or>. The key <and> always has precedence over <or> .
where / key	0 - 1	string	Type of the key.
where / operator	0 - 1	string	Operator
where / value	0 - 1	string	Key value
key	0 - 1	string	Inquiring the values of a special key; possible keys depend on the task, e.g.: <ul style="list-style-type: none"> • owner • name • id • created
order			Sorting
order / key			Keyword by which the data are to be sorted
order / mode	0 - 1	enum	Sort sequence asc = ascending desc = descending
history	1		History Inquire
history / id	0 - 1	string	For single inquire. ID of the object. The ID will be created by the system when carrying out a list inquire.

17.2 History Inquire Tasks (0713)

With this task you can inquire all log entries of the registry operations. It includes among others the NIC status of your requests.

With the single inquire you receive the data of a single request. With the list inquire you receive the data of several requests.

This table contains all values that are returned for the Info and List Tasks by default or respectively returned by using a keyword.

Description of the columns:

- **Value:** name of the returned value
- **Description:** what does this value mean.
- **Info:** the value is returned by default by the **Info** task
- **Info <key>:** The value can be inquired additionally with an **Info** task by using the <key> key.
- **List:** The value is returned by default by the **List** task
- **List <key>:** The value can be inquired additionally with a **List** task by using the <key> key.
- **Order:** Sorting
- **Where:** Filtering

Value	Description	Info	Info <key>	List	List <key>	order	where
id	Entry ID in the request log	yes		yes		yes	yes
owner	Object owner	yes		yes		yes	yes
object	Objektyp, z. B. domain	yes			yes	yes	yes
type	Object type, e.g. domain, zone, ...	yes		yes		yes	yes
status	Status after processing	yes		yes		yes	yes
created	Creation date of the request.	yes		yes		yes	yes
action	Request type	yes		yes		yes	yes
function_code	Function code of the request.	yes		yes		yes	yes

Examples for XML Requests and Answers

Example Request: list inquiry (all objects beginning with "e")

```
<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0713</code>
```

```

    <view>
      <offset>0</offset>
      <limit>30</limit>
      <children>0</children>
    </view>
    <where>
      <key>object</key>
      <operator>like</operator>
      <value>e*</value>
    </where>
  </task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <history>
        <action>create</action>
        <created>2006-05-23 14:04:27</created>
        <function_code>0101</function_code>
        <id>43628</id>
        <object>example.com</object>
        <owner>
          <context></context>
          <user>customer</user>
        </owner>
        <status>error</status>
        <type>domain</type>
      </history>
      <history>
        <action>create</action>
        <created>2008-12-03 17:30:11</created>
        <function_code>0101</function_code>
        <id>1804478</id>
        <object>authinfo.de</object>
        <owner>
          <context></context>
          <user>example2</user>
        </owner>
        <status>success</status>
        <type>domain</type>
      </history>
      <summary>l2</summary>
    </data>
    <status>
      <code>S0713</code>
      <text>History data was inquired successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

Example Request: list inquiry (all successfully processed transfer-ins)

```

<?xml version="1.0" encoding="utf-8"?>
<request>
  <auth>
    <user>customer</user>
    <password>password</password>
    <context></context>
  </auth>
  <task>
    <code>0713</code>
  </task>
  <view>

```

```

    <offset>0</offset>
    <limit>30</limit>
    <children>0</children>
  </view>
  <where>
    <key>function_code</key>
    <operator>like</operator>
    <value>0104001</value>
  </where>
</task>
</request>

```

Example Response

```

<?xml version="1.0" encoding="utf-8" ?>
<response>
  <result>
    <data>
      <history>
        <action>transfer_in_nack</action>
        <created>2008-11-26 15:35:45</created>
        <function_code>0104001</function_code>
        <id>1804454</id>
        <object>example.info</object>
        <owner>
          <context></context>
          <user>customer</user>
        </owner>
        <status>error</status>
        <type>domain</type>
      </history>
      <history>
        <action>transfer_in_ack</action>
        <created>2009-03-02 17:25:18</created>
        <function_code>0104001</function_code>
        <id>1809423</id>
        <object>example.eu</object>
        <owner>
          <context></context>
          <user>customer</user>
        </owner>
        <status>success</status>
        <type>domain</type>
      </history>
      <history>
        <action>transfer_in_nack</action>
        <created>2009-04-09 17:01:08</created>
        <function_code>0104001</function_code>
        <id>1810562</id>
        <object>example.at</object>
        <owner>
          <context></context>
          <user>customer</user>
        </owner>
        <status>error</status>
        <type>domain</type>
      </history>
      <summary>4</summary>
    </data>
    <status>
      <code>S0713</code>
      <text>History data was inquired successfully.</text>
      <type>success</type>
    </status>
  </result>
</response>

```

18 Addendum

18.1 Changelog

You can view the changes in the last versions of the documentation here.

Version	Date	Chapter/ Subject	Changes
16.0	06/2016	Description of the XML Keywords	New keyword overwrite_authinfo
		Additional Information for Domain Contacts (Extensions)	.barcelona extensions
		All Info- and List- tasks	Added tables listing detailed information about which values are returned by default when using the optional keywords. In addition, added information for every keyword stating if it is possible to filter and sort by given keyword.
		User Info (1301004)	Removed deprecated nic_response key. IP restrictions bloc added.
		Domains/ XML keys description	Information added for the keywords <dnssec> and <use_auto_dnssec>
		Polling	New chapter "Polling"
		User	Unser delete 0301003 dokumentiert
		Zones	Zone Bulk Update: key word remove_main_ip; searching and replacing the main-ip with the zone uodate bulk request.
15.0	12/2015	Domains XML keys description table	"domain / addon / update / mode changed to "domain / extension / mode" "domain / addon / price_class" changed to "domain / extension / price_class "
			Obsolete key words deleted (transit, disconnect)
		Domain contacts	Keywords "handle/ addon / update / mode" changed to "handle / extension / mode"
			"Protection" key; default changed from A to B
		Domain Cancelation	Keyword "remove_cancellation" replaced by "remove_cancelation".
		Domain contact extensions	.ro extensions added .cat extensions removed, because they are no longer required .pro extensions deleted, because they are no longer required .aero extenions removed, becaues they are no longer required .swiss extensions added
		Various	Task names actualized from "inquire" to"info: Domain Inquire, TransferOut Inquire, IRTP Inquire, Zone Inquire, BackupMX Inquire, MailProxy Inquire,Contact

Version	Date	Chapter/ Subject	Changes
			Inquire, Redirect Inquire, User Inquire
14.0	08/2015	Object User Assignment	Object user assignment tasks: key word <cancelation> added.
			Task 1308002 (object user assignment bulk) added.
		MailProxy	New chapter added to this documentation
		User	User copy task added.
		DNSSEC	New chapter added to this documentation
		Zone Inquire	Inquire keys or list task added
		Domain Inquire	Inquire keys for list task added
			Task "IRTP Inquire" deleted. Substituted by "IRTP Domain Info" and IRTP Domain List".
		Domain Create, Domain Update	key use_auto_dnssec added
		BackupMX	New chapter added to this documentation
		Whois Privacy	Added for the tasks 0101, 0102, 0104
13.1	02/2015	Domains /PremiumDomains	An own table for explaining the premium domain XML keys. Some new XML examples.
		Domain Contact	XML example for <i>ContactCommentUpdate</i> changed.
13.0	01/2015		
		Domains/ Premium Domains	Documentation for Tasks <i>Domain Premium List</i> (0165) and <i>Domain Premium Info</i> (0164) Key <i>price_class</i> for premium domain registration added in the XML key description overview.
		Contacts	New Task Contact Comment Update (0302)
		XML keys description/ XM_ examples	For DNSSEC: Correction of the keyword <i>key</i> ; changed to <i>publickey</i> .
		Default request	Note concerning microsoft.net and BOM
12.0	07/2014	Preregistrations	Description of preregistration statuses added.
		Contact Inquire	Keys actualized, that can be inquired for list inquire
		Domain Inquire	Keys actualized, that can be inquired for list inquire
		Domain Contact Delete	The note concerning bulk deletion of contact has been changed, because the bulk request is possible now.
		Domain Transfer	NACK reasons actualized
		Domain cancelation	Tasks <i>Domain Restore</i> und <i>Domain Restore List</i> added.
		Handel Inquire	Key "verification" added
11.1	/01/2014	Domain Preregistration	Description of ipr extension keys
		Zone Delete	Note concerning scheduled requests removed

Version	Date	Chapter/ Subject	Changes
		General Requests	Occurence added for "order / key"
		Domain Contacts, Extensions	Format of trademark_regdate corrected
		History inquire	Operator changed from "eq" to "like"
		Domain Transfer In	XML example for bulk request corrected
		General	Format "date time" unitized.
11.0	01/2014	All chapters	Restructuring. You can find the description off all XML keys at the beginning of the chapter .now.
		Domain Transfer	Task "domain transfer plus with ownerchange" added
		Preregistration	Actualization of the templates
		Domain contact verification	Neues Kapitel
10.0	12/2012	Domain Contacts	Domain contact extensions for .au.
		Domains	New: domain backorder
		AuthInfo1 Create	New XML examples
		Domain Create Plus and Domain Update Plus	New XML examples
		Domain Create	Documentation of key "ignore_whois" added
		Interfaces	Java Client added
		Zones	Actualization of the SOA-Level values
9.0	07/2012		
		Domain Contacts	Extensions .hk, .pro
		Whois Privacy	Added for the tasks 0101, 0102, 0104
		Domain Preregistration	Preregistration tasks added
		DomainTransfer	AuthInfo2 Create
		Introduction	List inquire: note concerning the keywords sld, subtlid, tld
		Domain Create, Domain Update, Domain Transfer, Domain Owner change Cancelation Delete , Domain Contacts Create	Process graphic added, polling example responses added
		Domain update	key word domain_addon_update_mode
		Contact update	key word handle_addon_update_mode
		Contact create	key word force_handle_create
		Redirect	Domain redirect type https added

Version	Date	Chapter/ Subject	Changes
8.0	02/2011	All chapters	System messages actualized
		Introduction	Note, concerning the rules for ctid
		Replies	Graphics updated
		Domain update	IPv6: keys and example code
		Domain update/domain create	DNSSEC added
		Domain transfer	Graphics updated
		Domain inquiries	List inquire keys (dnssec, domainsafe) added
		Cancelation	Note, concerning the deletion of cancelations
		Addendum	Changelog added

18.2 Formats

Abbreviation	Description
cc	Country code (ISO 3166 Country Code)
contact-id	A contact ID the system generates when a domain contact is generated.
datetime	(SQL-Format) „yyyy-MM-dd hh:mm:ss“
email	A valid email address (local-part@domain-part, e.g. user@example.com)
enum	List of valid entries (enumeration)
int	Integer
ip	IP format (000.000.000.000)
nsetry	example.com IN A 1.2.3.4 www.example.com < http://www.domain.com > IN A 1.2.3.4 example.com IN MX 10 mailserver.com
phone	+1-123-123
string	String of characters
tld	TopLevel Domain, see http://www.iana.org/domains/root/db/

18.3 CodeList

Domain Create 0101
Domain Create Plus 0101
Domain Update 0102
Domain Update Plus 0102
Domain Renew 0101003
Domain Ownerchange 0104010
Domain Delete 0103
Domain List 0105
Domain Premium List (0165)
Domain Premium Info (0164)

Domain Cancelation Create 0103101
Domain Cancelation Update 0103102
Domain Cancelation Delete 0103103
Domain Cancelation Info 0103104
Domain Cancelation History Info 0103124
Domain Cancelation History List 0103125
Domain Restore List 0105005
Domain Restore 0101005

DomainTransferIn 0104
Domain Transfer Plus 0104
Domain Transfer Plus with Ownerchange 0104010
Domain Status 0102002
DomainTransferOut 010600
TransferOutInquire 0106001
AuthInfo Create 0113001
AuthInfo Delete 0113002
AuthInfo2 Create 0113003
IRTP Domain Info 0114008
IRTP Domain Restart 0114002

Domain Prereg Create 0110001
Domain Prereg Create Plus 0110001
Domain Prereg Confirm 0110007
Domain Prereg Update 0110002
Domain Prereg Delete 0110003
Domain Prereg Info 0110004
Domain Prereg List 0110005
Prereg Config List 0110006

Backorder Create 0141
Backorder Delete 0143
Backorder User List 0145

Backorder List 0146

Zone Create 0201

Zone Update 0202

Zone Update (Bulk) 0202001

Zone Import 0204

Zone Delete 0203

Zone Inquire 0205

DNSSEC Sign 0120001

DNSSEC Key Rollover 0120002

DNSSEC Unsign 0120003

DNSSEC Info 0120004

DNSSEC List 0120005

BackupMX Create 0521

BackupMX Delete 0523

BackupMX Inquire 0524

Contact Create 0301

Contact Update 0302

Contact Delete 0303

Contact Inquire 0304

Contact Comment Update (0302004)

Contact Verification Create 0341

Contact Verificationinfo 0344

Contact Verification List 0345

Contact Verification Resend Email 0349

Redirect Create 0501

Redirect Update 0502

Redirect Delete 0503

Redirect Inquire 0504

MailProxy Create 0511

MailProxy Update 0512

MailProxy Delete 0513

MailProxy Inquire 0514

BackupMXCreate 0521

BackupMX Delete 0523

BackupMX Inquire 0524

User Create 1301001

User Update 1301002

User Inquire 1301004
 User_Profile_Update 1301014
 User Copy 1301016

Object User_Assignment 1308
 Object User_Assignment_All 1308001
 Object User_Assignment_Bulk 1308002

History Inquire 0713

18.4 User Statuses

Every user has a “status” (Master, Admin, Normal) and a “sub-status” (+, *, normal, clone). These status define if he is allowed to create subusers and also the number of subuser levels he is allowed to create.

The “clone” sub-status is an exception: An user with the “clone” sub-status is not allowed to have own objects. He always works with the objects of his parent. If a “clone” user creates new objects, his parent is the owner of them.

The table shows which kind of subusers is allowed for the different user status.

	+	*	Normal	Clone
Master	(= superuser) Master* Master Master (clone) Admin+ Admin* Admin Normal + Normal* Normal	Master (clone) Admin+ Admin* Admin Normal + Normal* Normal	Admin+ Admin* Admin Normal + Normal* Normal	<i>Has the same privileges as the parent.</i>
Admin	Admin* Admin Admin (clone) Normal + Normal* Normal	Admin (clone) Normal + Normal* Normal	Normal + Normal* Normal	<i>Has the same privileges as the parent.</i>
Normal	Normal* Normal Normal (clone)	Normal (clone)	<i>Has not the privilege to create a new user.</i>	<i>Has the same privileges as the parent.</i>

18.5 .HK: Industry Types

0: Not Selected

010100: Plastics, Petro-Chemicals, Chemicals - Plastics and Plastic Products

010200: Plastics, Petro-Chemicals, Chemicals - Rubber and Rubber Products

010300: Plastics, Petro-Chemicals, Chemicals - Fibre Materials and Products

010400: Plastics, Petro-Chemicals, Chemicals - Petroleum, Coal and Other Fuels

010500: Plastics, Petro-Chemicals, Chemicals - Chemicals and Chemical Products

020100: Metals, Machinery, Equipment - Metal Materials and Treatment

020200: Metals, Machinery, Equipment - Metal Products

020300: Metals, Machinery, Equipment - Industrial Machinery and Supplies

020400: Metals, Machinery, Equipment - Precision and Optical Equipment

020500: Metals, Machinery, Equipment - Moulds and Dies

030100: Printing, Paper, Publishing - Printing, Photocopying, Publishing

030200: Printing, Paper, Publishing - Paper, Paper Products

040100: Construction, Decoration, Environmental Engineering - Construction Contractors

040200: Construction, Decoration, Environmental Engineering - Construction Materials

040300: Construction, Decoration, Environmental Engineering - Decoration Materials

040400: Construction, Decoration, Environmental Engineering - Construction, Safety
Equipment and Supplies

040500: Construction, Decoration, Environmental Engineering - Decoration, Locksmiths,
Plumbing and Electrical Works

040600: Construction, Decoration, Environmental Engineering - Fire Protection Equipment and
Services

040700: Construction, Decoration, Environmental Engineering - Environmental Engineering,
Waste Reduction

050100: Textiles, Clothing and Accessories - Textiles, Fabric

050200: Textiles, Clothing and Accessories - Clothing

050300: Textiles, Clothing and Accessories - Uniforms, Special Clothing

050400: Textiles, Clothing and Accessories - Clothing Manufacturing Accessories

050500: Textiles, Clothing and Accessories - Clothing Processing and Equipment

050600: Textiles, Clothing and Accessories - Fur, Leather and Leather Goods

050700: Textiles, Clothing and Accessories - Handbags, Footwear, Optical Goods, Personal
Accessories

060100: Electronics, Electrical Appliances - Electronic Equipment and Supplies

060200: Electronics, Electrical Appliances - Electronic Parts and Components

060300: Electronics, Electrical Appliances - Electrical Appliances, Audio-Visual Equipment

070100: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Kitchenware, Tableware

070200: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Bedding

070300: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Bathroom, Cleaning Accessories

070400: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Household Goods

070500: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Wooden, Bamboo and Rattan
Goods

070600: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Home Furnishings, Arts and
Crafts

070700: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Watches, Clocks

070800: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Jewellery Accessories
070900: Houseware, Watches, Clocks, Jewellery, Toys, Gifts - Toys, Games, Gifts
080100: Business and Professional Services, Finance - Accounting, Legal Services
080200: Business and Professional Services, Finance - Advertising, Promotion Services
080300: Business and Professional Services, Finance - Consultancy Services
080400: Business and Professional Services, Finance - Translation, Design Services
080500: Business and Professional Services, Finance - Cleaning, Pest Control Services
080600: Business and Professional Services, Finance - Security Services
080700: Business and Professional Services, Finance - Trading, Business Services
080800: Business and Professional Services, Finance - Employment Services
080900: Business and Professional Services, Finance - Banking, Finance, Investment
081000: Business and Professional Services, Finance - Insurance
081100: Business and Professional Services, Finance - Property, Real Estate
090100: Transportation, Logistics - Land Transport, Motorcars
090200: Transportation, Logistics - Sea Transport, Boats
090300: Transportation, Logistics - Air Transport
090400: Transportation, Logistics - Moving, Warehousing, Courier and Logistics Services
090500: Transportation, Logistics - Freight Forwarding
100100: Office Equipment, Furniture, Stationery, Information Technology - Office, Commercial Equipment and Supplies
100200: Office Equipment, Furniture, Stationery, Information Technology - Office and Home Furniture
100300: Office Equipment, Furniture, Stationery, Information Technology - Stationery and Educational Supplies
100400: Office Equipment, Furniture, Stationery, Information Technology - Telecommunication Equipment and Services
100500: Office Equipment, Furniture, Stationery, Information Technology - Computers, Information Technology
110100: Food, Flowers, Fishing and Agriculture - Food Products and Supplies
110200: Food, Flowers, Fishing and Agriculture - Beverages, Tobacco
110300: Food, Flowers, Fishing and Agriculture - Restaurant Equipment and Supplies
110400: Food, Flowers, Fishing and Agriculture - Flowers, Artificial Flowers, Plants
110500: Food, Flowers, Fishing and Agriculture - Fishing
110600: Food, Flowers, Fishing and Agriculture - Agriculture
120100: Medical Services, Beauty, Social Services - Medicine and Herbal Products
120200: Medical Services, Beauty, Social Services - Medical and Therapeutic Services
120300: Medical Services, Beauty, Social Services - Medical Equipment and Supplies
120400: Medical Services, Beauty, Social Services - Beauty, Health
120500: Medical Services, Beauty, Social Services - Personal Services
120600: Medical Services, Beauty, Social Services - Organizations, Associations
120700: Medical Services, Beauty, Social Services - Information, Media
120800: Medical Services, Beauty, Social Services - Public Utilities
120900: Medical Services, Beauty, Social Services - Religion, Astrology, Funeral Services
130100: Culture, Education - Music, Arts
130200: Culture, Education - Learning Instruction and Training
130300: Culture, Education - Elementary Education

130400: Culture, Education - Tertiary Education, Other Education Services
130500: Culture, Education - Sporting Goods
130600: Culture, Education - Sporting, Recreational Facilities and Venues
130700: Culture, Education - Hobbies, Recreational Activities
130800: Culture, Education - Pets, Pets Services and Supplies
140101: Dining, Entertainment, Shopping, Travel - Restaurant Guide - Chinese
140102: Dining, Entertainment, Shopping, Travel - Restaurant Guide - Asian
140103: Dining, Entertainment, Shopping, Travel - Restaurant Guide - Western
140200: Dining, Entertainment, Shopping, Travel - Catering Services, Eateries
140300: Dining, Entertainment, Shopping, Travel - Entertainment Venues
140400: Dining, Entertainment, Shopping, Travel - Entertainment Production and Services
140500: Dining, Entertainment, Shopping, Travel - Entertainment Equipment and Facilities
140600: Dining, Entertainment, Shopping, Travel - Shopping Venues
140700: Dining, Entertainment, Shopping, Travel - Travel, Hotels and Accommodation