DENIC

Registration System for .de

Table of Contents

General Information 7
Contact Details 8
Member Chat 9
24-7-Service 10
Overview of the .de Registration System 14
Brief Description of the Fronts 18
Transaction IDs 19
Domain Status 20
Contact Status 22
Dates and Times 22
Letter Case for Data in Requests 23
Access Control Limit (ACL) 24
Request Counting 24
Throughput: 24
Blocking 24
Quotas and Registration Accounts 25
RRI Live and RRI Rush - Table Overview 27
Description of Front Ends 30
Realtime Registry Interface (RRI) 30
RRI - Introduction and Access Requirements 30
Transport Formats 31
Session Management 31
Security at the RRI 33
Access to the RRI 36
IP Allowlisting 37
RRI Web Client 40
Processes, Checks and Risk Assessment 43
NIS 2 Implementation - Overview 43
Collecting and Maintaining Registration Data 43
What's new? 43
Ensuring Accuracy and Completeness of the Data 43
Our Principles 43
Verification 44
Which data do I have to verify as a DENIC member? 44
Communicating Verification Information 44
Risk Assesment and Quarantine 46
Publication of Domain Registration Data 46
Syntax Check 47
Completeness Check 48
Table Contact Data Checks 49
Risk Assessment 56

Determination of Risk Level ... 56
Risk Levels and Domain Requests ... 56
Risk Levels and Related Communication ... 57
Table of Risk Levels ... 58
Verification ... 61
Identity verification – methods, verification and claims ... 63
Methods ... 63
Claims ... 65
Evidences ... 66
Verification - Formats, Types, Values ... 74
Structure and rules for the verification information block for contacts of the types PERSON and ORG ... 76
Rules for the value range ... 97
Mapping table for method names to value names ... 101
Claims, verifications and methods that can be combined ... 102
Copy and paste ... 108
Requests to the .de Registration System ... 113
expireWarning and expire Message Type ... 113
Data Formats Data Types and Value Ranges ... 116
Description of Data Formats (Key/Value / XML) ... 117
Error Messages from the .de Registration System ... 119
Data Types and Value Ranges for Data Fields ... 125
Login and Logout ... 135
Login Request ... 137
Logout Request ... 141
Replies to LOGIN andLOGOUT Requests ... 143
Error Messages in Relation with LOGIN and LOGOUT Requests ... 145
Contact Requests ... 146
Contact Data Object ... 149
Information on the Phone Number ... 160
Explanation of the URI-Template ... 168
contactCREATE Request ... 170
contactUPDATE Request ... 210
Replies to contactCREATE and contactUPDATE Requests ... 251
contactCHECK Request ... 254
Replies to contactCHECK Requests ... 257
contactINFO Request ... 260
Replies to contactINFO Requests ... 297
Error Messages Related to Contact Requests ... 303
Requests Related to the Domain Inventory ... 309
Domain Statuses pendingCreate and serverHold ... 310
Domain Data Object ... 312
domainCREATE Request ... 324

domainUPDATE Request ... 327
domainCHHOLDER Request ... 331
domainRESTORE Request ... 335
domainDELETE Request ... 338
domainTRANSIT Request ... 342
Replies to Domain CREATE, UPDATE, CHHOLDER, RESTORE, DELETE, TRANSIT Requests ... 349
domainCHECK Request ... 351
domainINFO Request ... 356
Replies to domainCHECK and domainINFO Requests ... 386
Error Messages Related to Domain Requests ... 390
Requests Related to Public Contact Details of the Registrar ... 400
Public Registrar Contact Details Data Object ... 401
regaccINFO Request ... 412
Reply to regaccINFO Requests ... 416
Error Messages Related to RegAcc Requests ... 422
Provider Change Requests ... 422
Chprov Data Object ... 423
domainCREATE-AUTHINFO1 Request ... 425
domainCREATE-AUTHINFO2 Request ... 429
domainDELETE-AUTHINFO1 Request ... 434
domainCHPROV Request ... 438
Replies to CREATE-AUTHINFO1, CREATE-AUTHINFO2, CHPROV ... 445
Error Messages Related to Provider Change Requests ... 447
Requests Related to the Message Queue ... 455
QUEUE-READ Request ... 459
QUEUE-READ Responses ... 461
QUEUE-DELETE Request ... 461
QUEUE-DELETE Response ... 463
Error Messages Related to QUEUE-READ and QUEUE-DELETE Requests ... 464
authInfo2Delete Message Type ... 466
authInfo2Notify Message Type ... 468
authInfoExpire Message Type ... 470
chprovAuthInfo Message Type ... 473
contactDelete Message Type ... 475
contactVerificationRequired Message Type ... 477
domainDelete Message Type ... 485
domainStatusUpdate Message Type ... 505
expireWarning and expire Message Type ... 522
Message Codes ... 526
Structure and Components of Message Codes ... 526
Parsing the error code in the responses ... 530

Frame Conditions for Registration 533
Notes on Legal Aspects 542
Annexes 547
.de Registry Lock 547
GnuPG and PGP 560
DNSSEC 578
Network Byte Order 582
Reading and Sending Data 583
List of Message Codes 594
contact CREATE and contact UPDATE 594
domain CREATE 595
domain CHHOLDER 596
domain CHPROV 597
domain RESTORE 598
List of Message Codes for Verification 616
Link Collection 630

Sub-Title1

01 General Information

General Information 7
Contact Details 8
Member Chat 9
24-7-Service 10

General Information

Overview

Introduction

In this chapter you will find basic information.

Contents

This chapter covers the following topics:

- About this Document
- Links to documentation
- Contact Details

Target group

This document is primarily intended for developers.

In this document you will find all information about the DENIC registration system, including, for example:

- the real-time registration interface "RRI" (including the RRI web interface),
- GnuPG in the context of DENIC use and
- all error messages of the .de registration system, as well as
- an explanation of the structure of the message coding.

Further DENIC documentation:

How tos

- **Setting up Accesses**

  This brochure is a quick guide that explains step by step how to set up important accesses to DENIC.

- **Information for Domain Investors**

  This brochure contains information that is relevant to achieve a high throughput of requests.

- **Information for international Members**

  International members who have experience with other registries receive an insight into the technical situation in particular through this brochure.

- **Information for Experts**

  Contents are information about the RAI, the ACL or the DENIC RRI client. The brochure is aimed at people who know DENIC very well technically and want to quickly read up specific numbers, parameters or syntactic information.

Manuals

- **RAI - Registrar Administration Interface**

  It contains information on self-administration of a registration account.

- **Nameserver Predelegation Check**

  Contents are descriptions of the requirements for name servers and zone data.

- **Operator Change for DNSSEC Domains**

  The document contains information on key and operator changes at DNSSEC.

- **Information Service**

  Described in "Information Service" are the information options and the format of accounting data.

Contact Details

Address

DENIC eG
Business Services
Theodor-Stern-Kai 1
60596 Frankfurt am Main
Germany

Phone, Fax und E-Mail

Kontaktinformation
Contact information

Fax: +49 69 27235-234

Service hours

Monday to Thursday 8:00 bis 18:00 CE(S)T,
Friday 8:00 bis 16:00 CE(S)T

Member Chat

The DENIC member chat is accessible via the Mattermost chat system. On the client side, Mattermost can either be used in Internet browsers or it is available as an app. On the server side, DENIC is operating a self-hosted Mattermost instance.

The chat system is available exclusively for exchange between DENIC and the members and among the members themselves.

Information is exchanged in public and in private group channels and between users via private direct messages. On the next page, we have compiled the available chat options in a table.

How do I get access to the member chat?

Send a request to Member Services at DBS (Kontaktinformation
Contact information) with the email address you would like to use for the chat system, followed by your first name and surname. You will then receive an invitation link to the chat system, which is valid for 48 hours.

Channel	Public	Private	Description	Rules
General	x	-	General channel for all members	This channel is used to make announcements, inform about maintenance work and outages, and for general issues.

Channel	Public	Private	Description	Rules
Technical Advisory Council	x	-	This channel is used for discussions between the Technical Advisory Council and the DENIC members.	-
DENIC-<member number>	-	x	This channel is a member channel that can be used by all employees of the member. Upon request by DBS it can be set up as a private channel for the communication between the member and DBS.	At the start of the chat, the name of the person participating from DBS is communicated. If the person from DBS that participates in the chat changes during the ongoing chat, the name of the new person is also communicated.
<Personal name> for direct messages	-	x	This channel is used for chats between individual employees at the DENIC member and DBS.	At the start of the chat, the name of the person participating from DBS is communicated. If the person from DBS that participates in the chat changes during the ongoing chat, the name of the new person is also communicated.

24-7-Service

DENIC's 24/7 service is made available free of charge to all members of the Cooperative.

Availability

How to contact us

Phone: +49 69 27235 299

E-Mail: sos[at]denic[dot]de

(Note: The subject must start with "SOS:")

Reporting errors and malfunctions

This service shall be used exclusively for notifying DENIC of any errors and malfunctions of the following central services of DENIC:

- Primary and secondary name servers for the ccTLD ".de",
- Mail servers,
- Web servers and
- whois servers, domain and contact registration system.

Known disturbances are published on the mailing list and at status.denic.de.

Cyber Attacks

In addition to notifications about malfunctions of central services of DENIC, you can also use this number if you are no longer able to change passwords to DENIC systems on your own due to a cyber attack.

To verify such reports, we use contact options known to us, such as the 24h emergency number you have stored!

Availability outside of normal service hours

Outside of normal service hours, DENIC's 24/7 service is available as follows:

- Monday to Thursday 18:00 – 8:00
- Friday 16:00 to Monday 8:00, and

- on official public holidays in the German federal state of Hesse as well as on 24 and 31 December.

When using DENIC's 24/7 service, please observe the applicable rules and regulations

1. The Member as well as the Member's staff shall have the right to report errors and malfunctions to DENIC staff outside of DENIC's normal service hours within the scope of DENIC's 24/7 service. Any errors or malfunctions will be diagnosed and fixed as quickly as possible upon receipt of the report.
2. The Member and their staff is aware that DENIC's 24/7 service is made available exclusively for the purpose of notifying DENIC of any errors and malfunctions of any of its following central services, notably the primary and secondary name servers for the .de ccTLD, mail servers, web servers, whois servers, domain and contact registration system.
3. The Member is aware that any other problems not specifically listed in point 2 or which concern DENIC's general operation do not fall within the scope of the 24/7 service and must continue to be reported to DENIC only via its usual telephone numbers and within its normal business hours, which is when they will also be dealt with. This applies, in particular, to questions concerning the status of requests, individual problems with requests, and so on.
4. Any DENIC employee assigned to the 24/7 service is required to keep a complete log of all incoming calls and to record the name of each caller and the member-ID. They are not required to act upon calls unless this information is disclosed.
5. More particularly, the Member enters into an express commitment not to pass DENIC's 24/7 service number on to any customer or any third party (such as a POP or a reseller).
6. The Member expressly agrees that any reports to DENIC's 24/7 service which are not specifically included in point 2 and which reach DENIC outside of its normal office hours will be separately charged to the Member.

Sub-Title1

02 Overview of the .de Registration System

Overview of the .de Registration System 14
- Brief Description of the Fronts 18
- Transaction IDs 19
- Domain Status 20
- Contact Status 22
- Dates and Times 22
- Letter Case for Data in Requests 23
- Access Control Limit (ACL) 24
  - Request Counting 24
  - Throughput: 24
  - Blocking 24
- Quotas and Registration Accounts 25
- RRI Live and RRI Rush - Table Overview 27

Overview of the .de Registration System

Introduction

In this chapter you will find an overview of the registration system and general information.

Contents

The chapter contains the topics:

- Overview of the registration system
- Brief description of the front ends
- c
- Domain and contact status
- Time specifications
- Notation for data in requests
- Contingents and further registration accounts and an
- Overview of RRI Live and RRI Rush

The Real-time Registry Interface (RRI)

DENIC's real-time registration interface RRI (and the RRI web interface) can be used to create and maintain contacts and domains automatically. RRI is the heart of DENIC's registration system.

In this document you will find all information about RRI and the web interface.

Access to the .de Registry System

Access to the DENIC registration system will only be granted after the member has fulfilled the relevant requirements:

- been admitted to the Cooperative,
- transferred at least one capital share,

- signed the Agreement about the Use of the Cooperative's Services,
- Attended the member training,
    - at least one staff member must have passed the Acceptance Test at the end of the training and
- the primary RegAcc profile for the basic RegAcc for RRI Live must be initially filled in by the member with the mandatory data.

New members in particular are recommended to test all processes and order types in the test environment first (whereby all order types must initially be tested in the live environment before activation). Use of the test environment is free of charge.

> **Notice**
>
> RRI Rush
> Especially for members whose business model is focused on the re-registration of deleted domains, DENIC offers an additional platform in the form of RRI Rush.
>
> **Please note:**
>
> - Every member has access to RRI Live (basic RegAcc) by default and can additionally purchase access to RRI Rush.
> - A RegAcc can operate in either RRI Live or RRI Rush.

Locations

The RRI is part of the cloud infrastructure at DENIC, the DENIC cloud, with two geographic connection points for the RRI in:

- Frankfurt and
- Amsterdam.

Members connect to the RRI with their clients to one of these two locations when they log in. The clients are usually automatically connected to the location that is geographically closer to them. Both locations are operated simultaneously (active-active operation). If one location fails, the requests and responses are taken over by the other location.

> **Notice**
>
> The concept of rolling out new software by operating one location passively and the other location actively, with a daily location switchover that meant a brief interruption of the RRI, is no longer used by DENIC.
>
> There is a fixed maintenance window, which is announced in announcements, including possible effects on services. In addition, a brief interruption of the RRIs may occur due to the continuous delivery process of software. More information about “Maintenance” can be found in the service catalog.

The whois Information Service

You can use the member-whois information service (whois.member.denic.de) to request information about domains, contacts, registration accounts and DENIC members.

Registrar Administration Interface (RAI)

The Registrar Administration Interface, RAI in short, is a web-based management tool with which you can edit registrar-related information (RegAcc profiles). You can also use RAI to change your token for two-factor authentication.

Detailed information on RAI can be found in the documentation "RAI - Registrar Administration Interface", How to Set Up a Two-Factor Authentication (2FA).

Lists

In the accounting area on the member website:

https://member.secure.denic.de/en/service/financial-issues/accounting-data you will find lists of all your .de and ENUM domains. On the first working day (Monday to Friday with the exception of 24.12., 31.12. and public holidays in Hesse, Germany) of each month, you will also find the monthly accounting data here.

Detailed information on the daily lists can be found in the documentation "Information Services".

> **Notice**
>
> If you require additional lists, please contact Business Services at dbs@denic.de.

Nameserver Predelegation Checks with NAST

With the NAST web interface (NAST = **NA**me**S**erver **T**ester) you can carry out a nameserver predelegation check, as is also done for .de domains in the RRI before delegating a domain. The name servers of a domain are subjected to various tests to ensure that they are configured correctly and that the domain can be delegated safely and easily.

At https://nast.denic.de/ you will find the web interface and the download package which contains the REST web service, a command-line interface and the source code.

> **Hint**
>
> On the member websites you will find extensive information on the various services, FAQs and much more!

Notifies by DENIC

Some notifies are generated automatically at a specific point in time and then sent (e.g. messages announcing the imminent Expire of a domain).

These messages are either e-mailed to the e-mail address recorded in the RegAcc profile (EmailTo) or deposited in the (RRI) message queue of the RegAcc.

If a message is sent by e-mail or stored in the message queue can be defined separately for each individual RegAcc via the keyword MsgFeed in the RegAcc profile.

Service Standards

DENIC has defined a service catalog in order to make the services and their degree of fulfillment transparent, comprehensible and comparable. The Service Standards (https://member.secure.denic.de/en/service/downloads-and-archives) serve as a basis for the continuous optimization of technical and organizational processes.

Brief Description of the Fronts

Introduction

DENIC's registration system is based on the Realtime Registry Interface RRI developed by DENIC. You can process requests through your own RRI client or through the RRI web client made available by DENIC.

RRI

DENIC only provides the server. Any user has to develop their own client, which must be able to use the RRI protocol.

> **Hint**
>
> You will find an RRI toolkit on the DENIC member website to support you in implementing an RRI client.

RRI Advantages

- Very well suited for large job volumes
- Requests and responses in realtime
- Very quick

- Supports transport protocol TCP
- Makes available two request formats, key/value and XML
- Communication is TLS-secured

RRI Disadvantages

Users need know-how for developing their own clients.

RRI Webclient

You can use the Web Interface - and the included technical web client on the member website - to transfer requests to RRI via a TLS-secured connection.

> **Notice**
>
> If you have an RRI login you can use the web interface in parallel. Log in with your valid RRI login.

RRI-Web Client Advantages

- Suited for small job volumes
- "Mobile" use possible; Browser is installed on nearly every computer
- The technical web client can be used to validate the syntax of requests in both the key/-value and the XML format
- Rarely used request types (such as AuthInfo2) can be used with the web client
- Communication is TLS-secured

RRI Web Client Disadvantages

- No automated use possible
- Responses are provided exclusively via the web and cannot be viewed by the user (via web client) at a later date

Transaction IDs

STID

The Server Transaction ID (STID) ensures that the request from the RegAcc and the pertinent response of the RRI server can be referenced globally and uniquely for data writing requests. The STID is assigned by the RRI.

CTID

The Client Transaction ID (CTID) is optional and is assigned by the client (RegAcc) for all write-requests

The CTID must be unique and 3 to 64 characters long. The CTID corresponds to a sequence of characters chosen by RegAcc itself and may only be used once per RegAcc within 24 months. If the CTID is used again within the 24 months, the order will be rejected with an error.

You can use the CTID to assign requests and replies to each other. The following rule applies: The CTID is only included in the reply from the server if the .de registration system was able to parse the request.

Domain Status

A .de domain can have the following statuses:

free

The domain is not registered.

The only available request for a domain with the status “free” is CREATE.

connect

Example domain: denic-connect-example.de

The domain is registered and connected.

For a domain of your own inventory the following requests are available:

UPDATE, CHHOLDER, CREATE-AUHTINFO1, DELETE-AUHTINFO1, CREATE-AUTHINFO2, TRANSIT and DELETE.

For a domain of another than your own inventory the following requests are available:

CREATE-AUTHINFO2, CHPROV.

failed

Example domain: denic-failed-example.de

The domain was registered or restored (RESTORE request), but is not connected, because the Name Server Predelegation Check failed during domain registration or restoring. The domain must be connected within 30 days by means of an UPDATE or CHHOLDER request.

A domain with a “failed” status is not included in the .de zone.

For a domain of your own inventory the following requests are available:

UPDATE, CHHOLDER, CREATE-AUHTINFO1, DELETE-AUHTINFO1, CREATE-AUTHINFO2, TRANSIT and DELETE.

For a domain of another than your own inventory the following requests are available:

CREATE-AUTHINFO2, CHPROV.

invalid

Example domain: denic-invalid-example.de

The domain is invalid and cannot be registered

redemptionPeriod

Example domain: denic-rgp-example.de

A domain was deleted and is in the Redemption Grace Period for 30 days from the day of deletion.

A domain with a “redemptionPeriod” status is not included in the .de zone.

If the domain was part of your own inventory, you can restore it during the Redemption Grace Period by means of a RESTORE request.

Any provider can restore the domain by means of a CHPROV request with AUTHINFO2.

> **Notice**
>
> After the Redemption Grace Period has expired, domains are released randomly and uniformly on the following calendar day in a configurable time window (DENIC reserves the right to change the time window at any time).
>
> **Example:**
>
> Redemption Grace Period = 30 days / time window 02:00 a.m. to 04:00 a.m.
> August 1: DELETE
> August 31: End of the Redemption Grace Period
> September 1: Release domain randomly between 02:00 a.m. and 04:00 a.m.

Contact Status

A DENIC handle can have the following statuses:

free

The Contact does not exist.

exist

The Contact exists.

Dates and Times

Format

Dates and times are displayed as stipulated by the ISO Standard 8601. They are indicated in UTC.

- Dates are written: YYYY-MM-DD. Example: 2021-04-07
- Then follows T (for time) as a separator between date and time Example: 2021-04-07T
- Times are written: hh:mm:ss Example: 2021-04-07T13:16:00
- Then follows recommended information, the difference to the coordinated universal time (UTC). It is written: +hh:mm Example: 2021-04-07T13:16:00+01:00

DENIC uses this format for all services. Thus, also times and dates in messages of INFO and whois outputs etc. are displayed in this format

Time Servers of DENIC

DENIC operates two time servers that can be used via NTP (Network Time Protocol (Version 3)) for synchronization. Only NTP version 3 is supported:

ntp1.denic.de

ntp2.denic.de

A server with realtime clock gives the servers the actual time. If this time source is not available due to failure, the servers synchronize with the NTP server of the metrology institute Physikalisch-Technische Bundesanstalt Braunschweig.

Letter Case for Data in Requests

General Information

You can state all data of a domain request in case insensitive form.

Conversion of Specific Data

Some data is stored at DENIC in the following notation and output via the info services according to the table.

Date	Format	Example
DENIC handle	Uppercase	DENIC-1000002-DENIC
Domain	Lowercase	de-example.de

Date	Format	Example
Name server	Lowercase	ns1.de-example.de
NS-Enty	Host name Lowercase, Typ Uppercase	de-example.de IN A 81.91.170.12

Access Control Limit (ACL)

Explanation:

To ensure availability for all and thus equal opportunities, the RRI has an access control limit.

Request Counting

Every request is counted, whether valid or invalid.

Throughput:

- **RRI-Live, 04:00 - 02:00 CET:** 2000 requests / 60 seconds
- **RRI-Live, 02:00 - 04:00 CET aka purge phase:** 200 requests / 60 seconds
- **RRI-Rush, 00:00 - 24:00 CET aka one full calendar day;** 2000 requests / 60 seconds

Blocking

The number of requests must not exceed the limit in the defined time interval.

If the limit is exceeded, `error code 55000000002` issued.

- **If the limit is exceeded:** 120 seconds blocked
    - **Unblocking:** The request volume was below the maximum in the last 120 seconds (currently the limits are 2000 or 200 requests).

- **Continued blocking for 120:** The request volume was above the maximum in the last 60 seconds.

Quotas and Registration Accounts

Quotas and Quota Points

- Requests submitted to RRI are subject to a quota.
- At present, each time the RegAcc submits any type of RRI request, the quota is reduced by one point.
  - Example: 50,000 requests = 50,000 quota points
- Differentiation according to requests is technically possible. DENIC reserves the right to amend processing at any time.

> **Notice**
>
> **No quota points will be charged for the following requests**
>
> - LOGIN
> - LOGOUT
> - QUEUE-READ
> - QUEUE-DELETE

Basic Registration Account

RRI Live

- Every member of the Cooperative has one Basic Registration Account (RegAcc) with a monthly basic quota. Use of the basic quota does not incur any additional costs.
- The number of requests per day is calculated as follows:

Domain portfolio at the beginning of the month below or equal to 50,000 domains	Domain portfolio at the beginning of the month with more than 50,000 domains
50,000 requests per day	Requests per day = domain portfolio * 2

- Unlimited numbers of additional quotas, each comprising 50,000 requests per day, can be booked via DBS against payment.
- Additional quotas have a term of six months.

Additional Registration Accounts

RRI Live

- You can use 9 additional registration accounts for the RRI Live. Each account is assigned a monthly quota:

  Requests = 2 x domain inventory

- The 9 additional registration accounts for the RRI Live are subject to payment and can be booked via DBS.
- Additional quotas of 50,000 requests per day are subject to payment and can be booked for the additional registration accounts via DBS with no restriction on quota numbers.
- Additional quotas have a term of six months.

RRI Rush

- Additional registration accounts for RRI Rush can be set up by DBS. There is no restriction on numbers.
- A quota for an additional account in the RRI Rush environment comprises 50,000 requests per day. It is can be booked with DBS against payment.

Booking or Terminating Additional Quotas and Additional Registration Accounts

- With the additional quota, you can either access RRI Live or RRI Rush.
- To book additional quotas and additional registration accounts you must contact DBS by post or (master-key-signed) e-mail.

> **Hint**
>
> **Do you have any questions about the quota increase?**
>
> - Then please send us an email at dbs@denic.de.

- Your order must be on hand no later than five working days (Monday to Friday with the exception of 24.12., 31.12. and public holidays in Hesse, Germany) before the first day of the month in which you want to start using your additional quota or RegAcc.
- Provided the requested quota or account is available, it will be released for use as of the start of the month following the order or as of the month stated in the order.
- The additional quota will be charged on a monthly basis.
- Members and DENIC can abandon additional quotas and RegAccs after a minimum period of 6 months by written termination or by master-key-signed e-mail subject to one month notice to the end of the month.

Information about Quota Exhaustion

Whenever you submit an RRI request, DENIC will check the percentage of the RegAcc's quota already used up. DENIC informs in 3 steps about quota exhaustion: 60, 90 and 100% exhaustion.

RRI Live and RRI Rush - Table Overview

Notice on the Throughput of Requests

The ACL controls the throughput of the requests at the various locations. The throughput per registration account (RegAcc) is determined by a time interval and the environment to which the request is sent. See table below.

Differences

	RRI Live	RRI Rush
Basis RegAcc	Yes	No
RRI Web Client	Yes	No
Maxum number of additional RegAccs	9	Unlimited
Basic quota	Basic RegAcc: 2 x domain inventory, at least 50,000 requests per day	No
	RegAcc 2-10: 2 x domain inventory	
Additional quota	50,000 requests per day per RegAcc	50,000 requests per day per RegAcc
Throughput of requests	2:00 a.m.– 4:00 a.m. CE(S)T	
	200 requests / minute per RegAcc	2,000 requests / minute per RegAcc
	4:00 a.m.– 2:00 a.m. CE(S)T	
	2,000 requests / minute per RegAcc	2,000 requests / minute per RegAcc
Sessions per location	7 per RegAcc	7 per RegAcc

Sub-Title1

03 Description of Front Ends

Description of Front Ends 30
Realtime Registry Interface (RRI) 30
RRI - Introduction and Access Requirements 30
Transport Formats 31
Session Management 31
Security at the RRI 33
Access to the RRI 36
IP Allowlisting 37
RRI Web Client 40

Description of Front Ends

Contents

This chapter describes the front ends of the .de registration system.

Inhalt

This chapter comprises the following sections:

- Realtime Registry Interface
- RRI Web Client

Realtime Registry Interface (RRI)

Introduction

In this section you will find basic information about the Realtime Registry Interface (RRI).

Contents

This section covers the following topics:

- RRI: Introduction and Access Requirements
- Transport Formats
- Session Management
- Security
- Access to RRI

RRI - Introduction and Access Requirements

Why RRI?

The RRI interface, which was developed in 2004, offers all the advantages of a realtime registration interface and is able to map all data and processes used at DENIC.

EPP (Extensible Provisioning Protocol (EPP) Transport over TCP) is the standard developed by the Internet Engineering Task Force (IETF) to enable realtime communication between registries and registrars. This standard, in particular the early implementations of it, were not suited for the applications of DENIC. However, the RRI of DENIC includes many elements of the IETF standard.

Access Requirements

The realtime registration system is based on a client/server model. To be allowed access you need to have established a RegAcc profile containing at least all mandatory data and an RRI login.

Transaction Model

A transaction comprises a request and the pertinent response. Competing operations intended for one and the same object will be rejected with an error message.

Transport Formats

RRI supports the transport protocol TCP.

Session Management

Introduction

To set up a session, you must execute a LOGIN command. You must do this before you transmit any other RRI commands. This establishes your identity and determines the authorization information for the whole duration of the session.

To log out, clients should use the LOGOUT command. When it receives a LOGOUT command, the server terminates the session. It is not possible to send any further RRI commands to the server after that.

The following rules apply for the session management:

- You may hold 7 parallel sessions for each RegAcc.
- If the maximum number of sessions is reached, it is impossible to set up any further sessions. They will be rejected with an error message.

DENIC reserves the right to amend processing accordingly at any time.

Error Messages Related to Session Management

Below we have listed the error messages you may be confronted with during session management.

Error Code	Error Message	Brief Description
83000000000	Request message too long	The PDU is too large. The size of the PDU must not exceed 64KB.

Security at the RRI

Security

DENIC uses TLS (RFC 6125) to safeguard the Realtime Registry Interface. By applying this tool, all the requirements concerning authentication of server, message integrity and confidentiality through encryption are met.

For server authentication vis-à-vis the client you can use an X.509-version-3 certificate.

Client authentication vis-à-vis the server is made by means of the user and the password.

Validation of Server Certificate

When communicating with DENIC's RRI server you must always use a TSL-encrypted connection. With the certificate used for this purpose you may also check if you are indeed connected with no other but DENIC's RRI server. Three different alternatives are available for validation.

- **No validation:** Your are not in any way forced to check/validate the RRI server certificate. However, if you do not carry out any validation, you will never have any guarantee that you are really and definitively connected with DENIC's RRI server. Performing no validation represents a security risk, so DENIC does not recommend this method!

- **Validation via the Root-CA, this is the method that DENIC recommends:** When validation is executed by the root certification authority (Root-CA), a check is performed to establish which certification authority signed the RRI server certificate. Currently, the RRI server certificate is signed by a CA which in turn has been signed by several superordinate certification bodies. So you should process requests as follows:
  - Initialize TLS
  - Read in the root certificate, if necessary (but note that the root certificate is not currently contained in all libraries - for example not in OpenSSL).
  - Open TPC-Connect
  - Start TLS
  - Check certificate (RFC 6125)
  - Send your request
  - Read the response
  - Close the connection
  - End TLS

    The advantage of this method is that, when DENIC renews its certificate, there is usually no need for the client to take any action at all, even if there is a change in the certification authority.

Transport encryption

The RRI server offers the following cipher suites and TLS protocols in all environments (live and test):

TLS Protocol Version	Cipher Suites
1.3	128 Bit / TLS_AES_128_GCM_SHA256 Curve 25519 DHE 253 (preferred)
1.2	256 Bit / ECDHE-RSA-AES256-GCM-SHA384 Curve 25519 DHE 253 (preferred)
1.2	256 Bit / DHE-RSA-AES256-GCM-SHA384 DHE 1024 bits
1.2	256 Bit / ECDHE-RSA-AES256-SHA384 Curve 25519 DHE 253
1.2	256 Bit / DHE-RSA-AES256-SHA256 DHE 1024 bits
1.2	128 Bit / AES128-SHA

Access to the RRI

Access

You may enter and change an RRI login for Live and Rush and the related password via the web application RAI (RegAcc Administration Interface).

RRI Login

Client authentication vis-à-vis the server is implemented by means of the user and the password (encrypted channel).

The value entered for "User" is the login in form of an existing DENIC handle of the RegAcc.

Password

You may use all visible Unicode characters (according to Unicode Version 3.1) as password.

The password is not transmitted to DENIC in plain text but as an MD5 hash. DENIC places at disposal a tool for creating the hash on the member website (Hash Generator). When you use a tool such as md5sum, keep in mind that the line break must not be taken into consideration for calculating the hash. (Example: "echo –n MyPW | md5sum" but not "echo MyPW | md5sum")

> **Notice**
>
> Choose a secure password.

Server Data

- Live & Rush environment

  Host name: rri.denic.de

  Port for TCP: 51131

- **Test environment**

Host name: rri.test.denic.de

Port for TCP: 51131

Particularities

- **Test environment**
  - RGP lasts one day
  - AuthInfo2 password is "secret"
  - Domain with status "failed" is deleted after one day
  - RGP ends between 10am and 12pm
  - ACL is 100 requests / minute

IP Allowlisting

> **Caution!**
>
> IP Allowlisting will become mandatory for all members starting February 3, 2026! From that point on, you will need to enter IP addresses.

> **Notice**
>
> You can enter your IP addresses for your RegAccs in the RAI before February 3, 2026, to activate or try out IP Allowlisting.

Members who exclusively use the RRI Web Client

Members who use the RRI web client do not need to set up IP allowlisting in the RAI. By logging in to the member page on the RRI web client, you have already authenticated yourself and do not need to set up an additional list of allowed IP numbers in the RAI.

Members who use the RRI and the RRI Web Client

Members who use a self-hosted RRI client and the RRI web client only need to enter IP addresses or networks for the RRI client in the RAI.

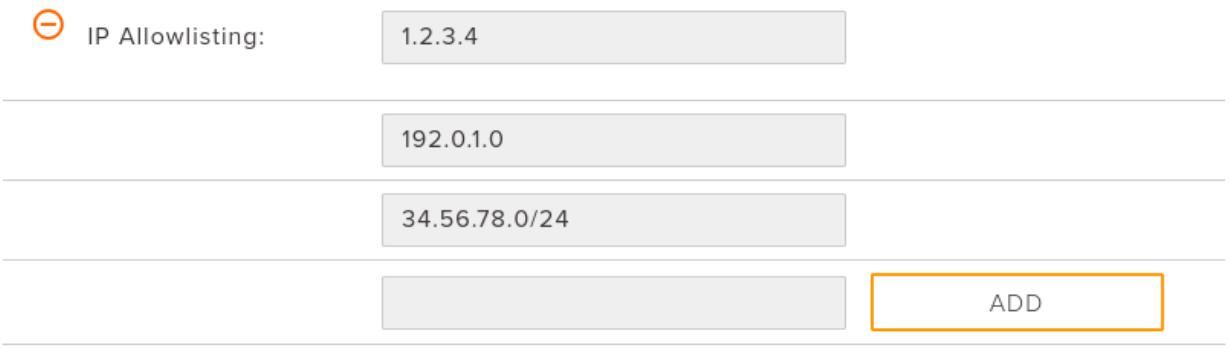
Setting up IP Allowlisting in the RAI for use with the RRI

> **Caution!**
>
> IPv6 is not supported!

In addition to the user and password for the RRI, the system checks whether the connection is being made via an authorized IP address. The IP address must be stored in the RAI beforehand.

- A maximum of 20 different entries can be stored.
- Address masks /32 to /24 are supported.
- The same networks may be used for multiple RegAccs.

Figure 1: Entry in the RAI



Hint

Please keep the following in mind when entering IP addresses in the RAI.

Assuming you want to enter a total of 5 IP addresses, it makes sense to enter all 5 IP addresses for the RegAcc at once and then save the change.

What would happen if you only entered one IP address first?

- IP allowlisting would be activated for this one IP address.
- However, this would also immediately block your 4 other IP addresses for RRI.

Summary

1. Remember that IP allowlisting is activated immediately as soon as you enter and save an address.
2. Think about how many IP addresses you want to enter beforehand.
3. Then enter all IP addresses at once and save your changes.

Additional tip

Of course, you can do the same if you want to deactivate IP allowlisting (only possible until February 3, 2026). Delete all IP addresses and then save your changes.

RRI Web Client

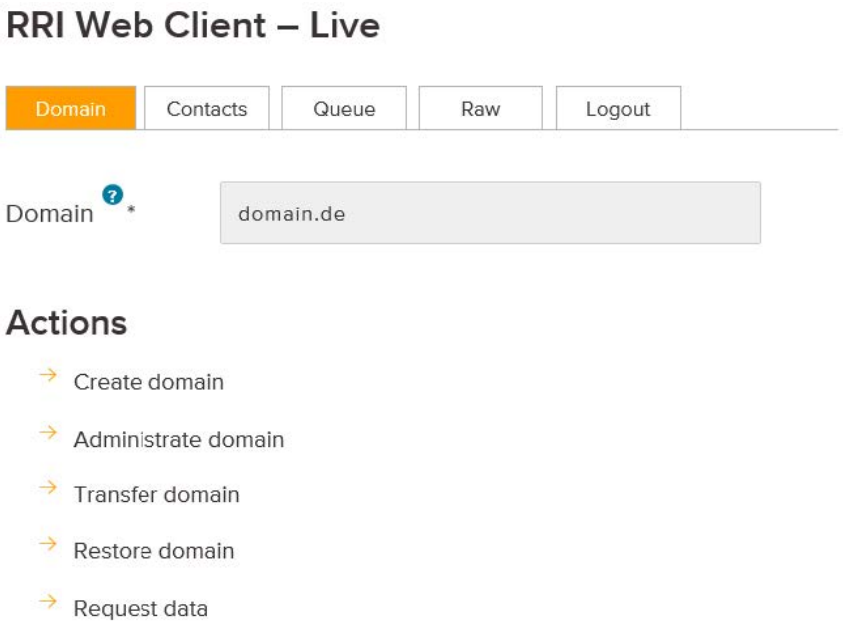
Web Interface

Functions

The web interface provides you with a simple solution for processing all request forms of .de web-supported and in real time (in both the test and the live environments).

Access

You will find the web interface at https://member.secure.denic.de/en/service/tools/rri-web-client/. For accessing, please use an RRI login that is valid for the respective environment. For selecting the request type you need, go to “DOMAIN”, “CONTACTS”, “QUEUE”:



Technical Web Client

Functions

The technical web client included in the web interface is located under the tab “Raw”. With the web client you can easily test RRI requests, their syntax and the RRI response pattern. You can enter RRI requests in the respective applicable data format (key/value or XML). The reply will have the same format as the request. This is default for RRI.

Sub-Title1

04 Processes, Checks and Risk Assessment

Processes, Checks and Risk Assessment 43
NIS 2 Implementation - Overview 43
Collecting and Maintaining Registration Data 43
What's new? 43
Ensuring Accuracy and Completeness of the Data 43
Our Principles 43
Verification 44
Which data do I have to verify as a DENIC member? 44
Communicating Verification Information 44
Risk Assesment and Quarantine 46
Publication of Domain Registration Data 46
Syntax Check 47
Completeness Check 48
Table Contact Data Checks 49
Risk Assessment 56
Determination of Risk Level 56
Risk Levels and Domain Requests 56
Risk Levels and Related Communication 57
Table of Risk Levels 58

Processes, Checks and Risk Assessment

NIS 2 Implementation - Overview

Collecting and Maintaining Registration Data

What's new?

- Due to the requirements of NIS 2, we will record a phone number of the domain holder in addition to the other data.

> **Caution!**
>
> Members have one year to update existing Holder handles with a phone number!

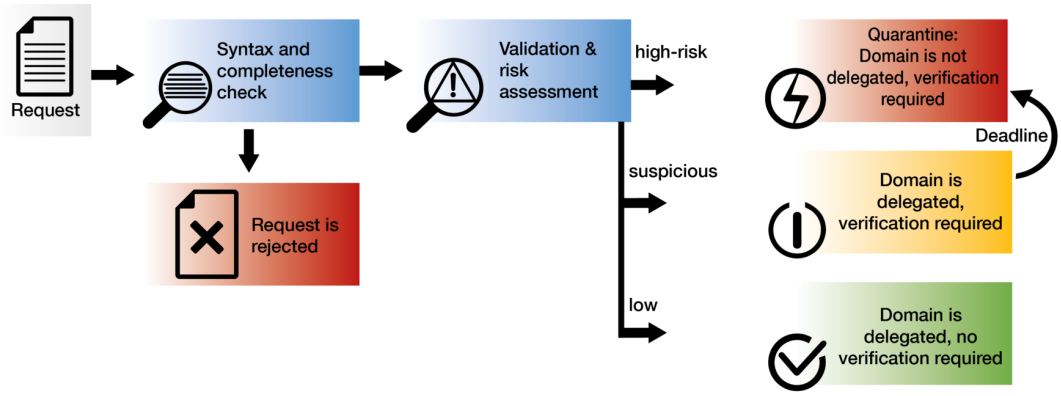
- We will publish the following data via the information services:
  - Legal person (domain holder of “ORG” type)
  - Domain registration date
  - Administering DENIC member
  - at a later date: Administering RegAcc

Ensuring Accuracy and Completeness of the Data

Our Principles

- Future-oriented, scalable in a flexible, risk-based approach
- Regular check only for new / updated / transferred domains
- Check of domains already in our inventory only in case of complaints or irregularities

- Ex post and ex ante verification possible



Verification

Which data do I have to verify as a DENIC member?

- Name and address
- E-mail address (analogous to gTLD process)

Communicating Verification Information

Once the verification by the member has been completed successfully, an update must be performed for the relevant holder handle:

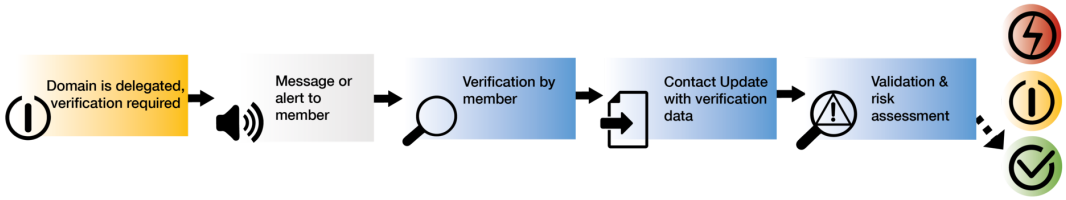
- With updated holder data, if necessary
- With meta information about the verification process conducted by the member
    - Verified claims (name, address, e-Mail address)
    - Time stamp
    - Audit reference (ticket ID etc.)
    - Verification method used
    - Information about the verified evidence
    - Trust framework

- Verification result (success or failed)

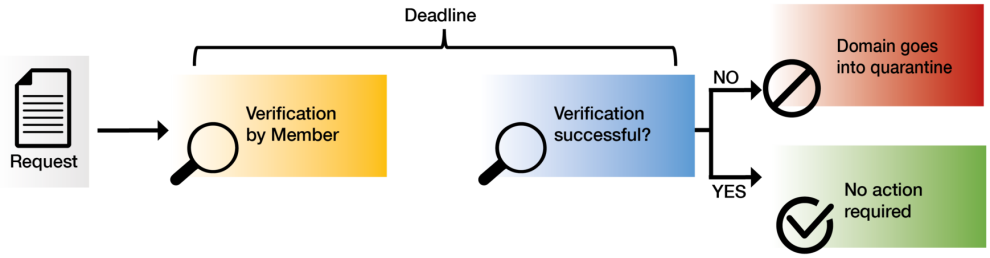
> **Caution!**
>
> If verification fails, the domain is always quarantined!

- A reference to a previous verification can be used
- Meta information can be provided directly with a Contact Create or a Contact Update to show that an ex-ante verification was performed.
- DENIC permits a wide range of evidence and methods. The member is free to choose which of these it wants to support.
- DENIC makes available process mocks in the test environment which members can use to test and implement the verification processes in their own systems.

Verification of Name and Address



Verification of E-Mail Address

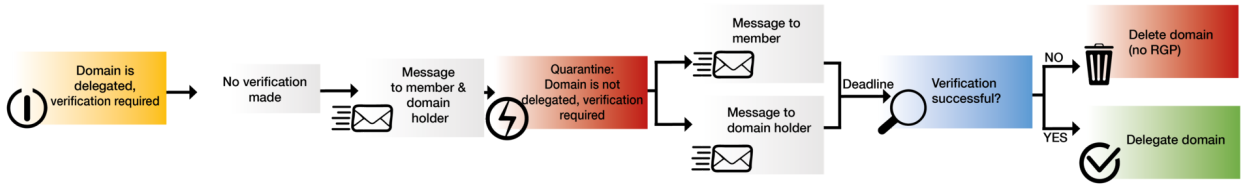


Risk Assesment and Quarantine

For new (Domain CREATE), updated (Domain CHHOLDER and Contact UPDATE) and transferred (Domain CHPROV) domains, DENIC asynchronously carries out a risk assessment.

DENIC applies a traffic light system for the risk assessment:

- Risk level "low" – The domain is registered and delegated.
- Risk level "suspicious" – The domain is registered and delegated, must however be verified within a defined period. If the domain is not verified, it is put into quarantine (risk level "high").
- Risk level "high" – The domain is registered but not delegated and immediately put into quarantine. If the domain is not verified within the defined period, the domain is deleted.



Publication of Domain Registration Data

The following data will be published via the information services:

- Legal persons (defined on the basis of the contact type "ORG"; Members must ensure assignment of correct type)
    - Name
    - Address
    - E-mail adress
    - Phone number
    - Domain registration date

- Administering DENIC member
- at a later date: administering RegAcc

- Natural persons (defined on the basis of the contact type "PERSON"; Members must ensure assignment of correct type)
  - Domain registration date
  - Administering DENIC member
  - at a later date: administering RegAcc

Members have the obligation to inform the domain holder once a year about the holder data and their potential publication in the whois.

Members must effectively inform end customers about publication of data.

Syntax Check

The syntax check is one of the checks to which DENIC submits the data provided by a member in a request. Within the scope of these checks, various requests are checked that can be composed of contact objects, domain objects and other request parameters.

The checks are carried out for both request formats, i.e. key/value and XML. The permitted data are usually described in detail in tables in the documentation. They include the value range, permitted length and similar information. You find the details in the description of the requests under "Requests to the .de Registration System" on page 113.

Here is a list of the requests which are subjected to the syntax check:

- Contact CREATE Request,
- Contact UPDATE Request,
- "domainCREATE Request" on page 324,
- "domainCHHOLDER Request" on page 331,

- "domainCHPROV Request" on page 438 and
- "domainUPDATE Request" on page 327.

Completeness Check

Introduction

- During the completeness check, the contact data are checked on syntactic correctness.
- The check is conducted exclusively for contact CREATE and contact UPDATE requests.
- Prior to the check, superfluous blanks are deleted:
  - All blank spaces before the first and after the last string are deleted.
  - Only one blank space is permitted between the strings, all further blank spaces are deleted.

Overview

Table Contact Data Checks, overview of all checks with examples

Table Contact Data Checks

> **Notice**
>
> Special characters are all characters outside the value ranges of “alpha-val”, “idn-val” and “digit-val”, see table.

Type	Value Ranges	Definition
ace-val	%xx41-%x5A / %x61-%x7A / %x2D	A-Z / a-z / hyphen
alpha-val	%x41-%x5A / %x61-%x7A	A-Z / a-z
authinfo-val	%x41-%x48 %x4A-%x4E %x50-%x5A %x61-%x6B %x6D-%x6E %x70-%x7A / %x32-%x39 / %x2B / %x2D / %x2F / %x2A	A-H J-N P-Z a-k m-n p-z / 2-9 / plus sign / hyphen / slash / asterisk
digit-val	%x30-%x39	0-9
dnskeyval	%x41-%x5A / %x61-%7A / %x30-39 / %x2D / %x2E / %x2F / %x2B	A-Z / a-z / 0-9 / hyphen / dot / slash / plus sign

Type	Value Ranges	Definition
idn-val	%xE1 / %xE0 / %x103 / %xE2 / %xE5 / %xE4 / %xE3 / %x105 / %x101 / %xE6 / %x107 / %x109 / %x10D / %x10B / %xE7 / %x10F / %x111 / %xE9 / %xE8 / %x115 / %xEA / %x11B / %xEB / %x117 / %x119 / %x113 / %x11F / %x11D / %x121 / %x123 / %x125 / %x127 / %xED / %xEC / %x12D / %xEE / %xEF / %x129 / %x12F / %x12B / %x131 / %x135 / %x137 / %x13A / %x13E / %x13C / %x142 / %x144 / %x148 / %xF1 / %x146 / %x14B / %xF3 / %xF2 / %x14F / %xF4 / %xF6 / %x151 / %xF5 / %xF8 / %x14D / %x153 / %x138 / %x155 / %x159 / %x157 / %x15B / %x15D / %x161 / %x15F / %x165 / %x163 / %x167 / %xFA / %xF9 / %x16D / %xFB / %x16F / %x171 / %x169 / %x173 / %x16B / %x175 / %xFD / %x177 / %xFF / %x17A /17E / %x17C / %xF0 / %xFE / %xDF	áàăâåäãąāæćĉčċçďđéèĕêěëėęēğĝġģĥħíìĭîïĩįīıĵķĺľļłńňñņŋóòŏôöőõøōoeκŕřŗśŝšşťţŧúùŭûůüűũųūŵýŷÿźžżðþß
ip-val	%x30-%x39 / %x2E / %x3A	ASCII: 0-9 / dot / colon
name-val	%x41-%x5A / %x61-%x7A / %x30-%x39 / %x20 / %x26 / %x2D / %x2E / %xA0-%xFF	ASCII: A-Z / a-z / 0-9 / space / ampersand (@) / hyphen / dot / visible characters from Latin-1
nserver-val	%xx41-x5A / %x61-%x7A / %x30-%x39 / %x2D / %x2E / %x3A / %x20	ASCII: A-Z / a-z / 0-9 / hyphen / dot / colon/ space

What is checked?	not permitted	Example(s)	Error code	Error message(s)
Name	numbers only	123456	63400000001	The value for keyword "Name" has an invalid format
	special characters only	%%%%%%		
	less than 3 characters	D		
	multiple use of only one specific letter	DDDDDDDDDD DDDD DDDDDD DDD-D		
	Not permitted for the contact type "PERSON": Three-character string with one special character	1%2		

What is checked?	not permitted	Example(s)	Error code	Error message(s)
	Not permitted for the contact type "PERSON": at least 5 times the same character in a row	xxxxxx111111		

What is checked?	not permitted	Example(s)	Error code	Error message(s)
Address	Numbers only	123456	63400000002	The value for keyword "Address" has an invalid format
	special characters only	%%%%%%		
	less than 3 characters	D		
	multiple use of one specific letter only	DDDDDDDDDD		
	at least 5 times the same character in a row	xxxxxx111111		

What is checked?	not permitted	Example(s)	Error code	Error message(s)
Postal code	**Allowed for DE:** 5-digit and purely numerical (0-9)	60329	63400000003	**If the conditions for Germany (DE), Austria (AT) and Switzerland (CH) are not met, the error message:** `The value for keyword "PostalCode" has an invalid format`
	Allowed for AT: 4 digits and purely numerical (0-9)	6691		
	Allowed for CH: 4 digits and purely numerical (0-9)	8090		

What is checked?	not permitted	Example(s)	Error code	Error message(s)
Town or city	Numbers only	123456	63400000004	The value for keyword "Address" has an invalid format
	special characters only	%%%%%%		
	special characters only	D		
	multiple use of one specific letter only	DDDDDDDDDD DDDD DDDDDD DDD-D		
	at least 5 times the same character in a row	xxxxxx111111		

Risk Assessment

The domain data are assessed by means of analysis methods of different sources. The data are then assigned to risk levels ("Determination of Risk Level" below) in accordance with the score achieved.

- Risk assessments are carried out for the following requests:
  - "domainCREATE Request" on page 324,
  - "domainCHHOLDER Request" on page 331,
  - "domainCHPROV Request" on page 438 and
  - "domainUPDATE Request" on page 327.

Determination of Risk Level

On the basis of the score achieved in the risk assessment, one of the following three risk levels is determined:

- low,
- suspicious and
- high.

DENIC does not publish the score, the analysis method or the source data used to carry out the assessment.

If a successful verification has already been performed and DENIC has been informed accordingly, the score will automatically be set to **low**.

Risk Levels and Domain Requests

- A domain is validated when certain domain requests are submitted and is subsequently assigned a risk level.
- There are three risk levels: **low**, **suspicious** and **high**.
- For the risk levels **suspicious** and **high**, deadlines apply.
- The risk level of a domain is automatically raised from **suspicious** to **high**, if the deadline of the **suspicious** risk level has expired.

- When the risk level is **high**, the domain is de-delegated, provided it was previously delegated.
- The member can verify a domain by submitting corresponding evidence within the applicable deadlines.
- When the deadline of a **high** risk level expires, the domain contract will be terminated and the domain deleted.
- Domains that are deleted due to a **high** risk level, are not eligible for the RGP (Redemption Grace Period) phase and cannot be reactivated.
- Pre-verifications can also be carried out before a domain request is sent.
- If the pre-verifications of the data are successful, the domain is assessed as **low** risk when a domain request is submitted.
- If a domain request is submitted with handles, the contact data of which have not successfully passed the pre-verification, the domain will immediately be assessed as **high** risk.

Risk Levels and Related Communication

- When the domain has the risk level **suspicious**, the member and the domain holder are notified of the threatening De-Delegation.
- When the risk level is **high**, the member and the domain holder are informed that the domain was de-delegated.
- When the deadlines of the risk levels have expired and the member has not provided any valid verification up to then, the domain holder is informed that their domain was terminated.
- The member is informed via message queue or by e-mail about the arbitrary deletion of the domain during the purge phase.

Table of Risk Levels

The table below explains the various risk levels and their consequences.

Traffic light colour	Risk level	Domain	Verification deadline	Explanation
green	low	• registered • delegated	none	• The validation of the data resulted in a score in the "low risk" range. • All the verification results of the contact data of the domain holder submitted to DENIC by the member have the value **success**.
yellow	suspicious	• registered • delegated • is de-delegated after a defined period	14 Days[1]	• The data of the domain are suspicious because the risk assessment resulted in a corresponding score. • If a successful verification is carried out within the deadline, the domain is reclassified to low risk.

Traffic light colour	Risk level	Domain	Verification deadline	Explanation
red	high	• registered • not delegated or de-delegated	90 Days[1]	The domain is assessed as high risk, • because the risk assessment resulted in a score in the high risk range, • because the 14-day deadline has expired and no verification has been made, • because the verification **failed**.

Sub-Title1
05 Verification

Verification 61
Identity verification – methods, verification and claims 63
Methods 63
Claims 65
Evidences 66
Verification - Formats, Types, Values 74
Structure and rules for the verification information block for contacts of the types PERSON and ORG 76
Rules for the value range 97
Mapping table for method names to value names 101
Claims, verifications and methods that can be combined 102
Copy and paste 108

Verification

For contacts that contain data about persons or organizations, it is possible to store information that confirms that the accuracy of the data has been verified by the member.

The process of verification is called “verification”. The information that represents the result of the verification is the “verification information”. The verification information is stored in the contact together with the data of a person or organization, but is located in a separate section called the “verification block”.

As a brief example, let's look at the arrangement in a contact as it was transmitted via a K/V order.

```
[VerificationInformation]
VerifiedClaim: name
VerifiedClaim: address
VerificationResult: success
VerificationReference: ABC123/45GHT
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: idcard
VerificationMethod: auth
TrustFramework: de_denic
```

In the verification information block, which is the contiguous part under “[VerificationInformation]”, various claims (name, address, e-mail) can be verified. There is also information about the verification and procedure used to check the data and other details.

The verification information block looks similar for a contact that was created via an XML request.

```
<verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:type="verification:verificationInformationType">
  <verification:verifiedClaims>
    <verification:claim>name</verification:claim>
    <verification:claim>address</verification:claim>
  </verification:verifiedClaims>
```

```
6   <verification:verificationResult>success</veri-
    fication:verificationResult>
7   <verification:verificationReference>ABC123/45GHT</veri-
    fication:verificationReference>
8   <verification:verificationTimestamp>2023-11-11T15:36:21+02:00</veri-
    fication:verificationTimestamp>
9   <verification:verificationEvidence>idcard</veri-
    fication:verificationEvidence>
10  <verification:verificationMethod>auth</veri-
    fication:verificationMethod>
11  <verification:trustFramework>de_denic</verification:trustFramework>
12  </verification:verificationInformation>
```

- Verification information is part of the address contacts and can be added via a CONTACT CREATE request.
- The verification information can be changed or deleted via a CONTACT UPDATE request.
- Verification information can be added easily or must be added as needed.
    - This is the case when DENIC checks the plausibility of the holder handle, among other things, for certain domain requests and requests verification from the member if anything is unclear. If your domain has multiple owners (multi-holder domain), all contacts must be verified.
- Members are solely responsible for verification. However, DENIC can carry out an audit and obtain an explanation of how the verification is carried out.

More information

More about proofs, claims and methods and how to combine them.
Identity verification – methods, verification and claims

What is the format of XML and K/V? What values are possible? This topic will tell you more.
Verification - Formats, Types, Values

Identity verification – methods, verification and claims

If members wish to verify the identity of a person, there are a number of different procedures based on proof that has been issued by official bodies or organizations. However, DENIC only accepts a defined group of procedures and proofs.

Methods

eIDAS

eIDAS is the abbreviation for "**e**lectronic **ID**entification, **A**uthentication and trust **S**ervices" and is the designation for EU Regulation 910/2014. Behind this is the regulation for EU member states on legally secure digital proof of identity, digital signatures and EUid wallets (amendment from 2024). The EUid Wallet, or "European Wallet for Digital Identity", enables EU citizens to identify themselves to companies and state institutions by means of an EUid Wallet app and the eID ("electronic identification", in Germany often the electronic ID card) stored there, by disclosing the necessary information, so-called attributes (or claims), of the digital identity.

POSTIDENT

The identity check service of the German postal service "Deutsche Post" is called "POST IDENTifikation" or just POSTIDENT. It comprises various procedures for establishing identification, including, for example, identification via video chat or in a post office. In this procedure, a person is identified by means of a valid identity document (identity card or passport).

Transaction Records

Transaction records are data that document satisfactorily that a transaction (such as payment by direct debit, transfer, credit card) has taken place. Members who have access to such data due to payments by domain holders can use these data as evidence.

Electronic Documents and Paper Documents

This group comprises a wide range of documents that contain usable identification information. Documents of this type could be, for instance, an identity card, an excerpt from a population register, a residence permit, a driver's license or an electricity bill. The members must guarantee by suitable procedures that the data in the document can be verified.

Photo-Ident and Video-Ident

These are procedures made available by various providers that identify a person through biometric data (face, fingerprint) in case of Photo-Ident or through a guided process by a qualified verifier in case of Video-Ident. The person is identified by means of physical characteristics (face) and specific documents associated to this person.

Reachability

In procedures checking on the reachability of a person, the person concerned must actively acknowledge their data. To identify a person via their address, a document is sent by post. The addressee must confirm the receipt of this document. In the course of this confirmation, the personal data are checked. An e-mail address can be checked by including a confirmation link in the mail. The recipient must click this link to confirm the address.

Claims

Claims are certain characteristics in a contact that can be verified.

DENIC accepts the verification of the

- name,
- address and
- e-mail.

Evidences

The table contains a list of the sources or certificates from which the claims may come. There may be requirements for the certificate that must be met in order for the certificate to be used. These may be, for example, certain organizations, bodies or authorities that issue and authenticate a certificate.

DENIC only accepts documentary evidence that meets these requirements.

The table also shows that the member has the option of offering the domain owner several procedures per documentary evidence.

Proof	Proof description	Method	Claims
ID card	Verification by an official and valid ID card; including emergency ID cards, ID cards for refugees, asylum seekers and apatrids	• eIDAS • POSTIDENT • electronic documents • paper documents • photo-ident • video-ident	• name • address (if available)

Proof	Proof description	Method	Claims
Passport	Verification by an official and valid passport	• POSTIDENT • electronic documents • paper documents • photo-ident • video-ident	name
Population register	Verification through an official excerpt from the population register (“Meldebescheinigung”) – (in countries where applicable)	• electronic documents • paper documents • photo-ident • video-ident	• name • address

Proof	Proof description	Method	Claims
Residence permit	Verification through an official and valid residence permit issued by an official authority	• electronic documents • paper documents • photo-ident • video-ident	• name • address
Proof of arrival	Verification through an official and valid proof of arrival issued by an official authority	• electronic documents • paper documents • photo-ident • video-ident	name

Proof	Proof description	Method	Claims
FührerscheinDriver's license	Verification through an official and valid driver's license issued by an official authority	• electronic documents • paper documents • photo-ident • video-ident	name
Company register	Verification through an excerpt from the commercial or cooperative register or a comparable official register or directory	electronic documents	• name • address
Founding documents	Verification through founding documents or equivalent evidentiary documents	• electronic documents • paper documents	• name • address
Federal Gazette	Verification through an excerpt from the Federal Gazette (“Bundesanzeiger”)	electronic documents	name

Proof	Proof description	Method	Claims
Record of payment via direct debit / transfer or credit card	Successful execution of a payment in which the data to be verified is securely transmitted by the payment service provider	transaction records	name
Record of payment via online payment	Successful execution of a payment in which the data to be verified is securely transmitted by the payment service provider	transaction records	• name • address • e-mail
Utility bill	Verification via a utility bill	• electronic documents • paper documents	• name • address
Bank statement	Verification via an official statement from a recognized banking institution	• electronic documents • paper documents	• name • address

Proof	Proof description	Method	Claims
Tax statement	Official certificate from the tax authorities of a country	• electronic documents • paper documents	• name • address
Written confirmation by an already verified natural person	A written/printed statement/letter from a recognised person or authority regarding the identity of the user	paper documents	• name • address • e-mail
Digital confirmation by an already verified natural person	A digital statement/letter from a recognised person or authority regarding the identity of the user	electronic documents	• name • address • e-mail
Postal transaction log	A digital transaction log of an appropriate confirmation action of postal verification, such as confirmation of receipt, entering a code or clicking a link	• POSTIDENT • reachability	• name • address

Proof	Proof description	Method	Claims
E-mail transaction log	A digital transaction log of an appropriate confirmation action of email verification, such as entering a code or clicking a link	reachability	e-mail
Address Database	Information from a reliable address database	transaction records	address

More information

What is verification? This topic gives you some basics to get you started.

"Verification" on page 61

What is the format of XML and K/V? What values are possible? This topic will tell you more.

Verification - Formats, Types, Values

Verification - Formats, Types, Values

A verification information block contains details about the verification. As with the other data for a contact (or domain), it is preceded by a keyword and followed by a valid value. The following tables show what can be combined in which form.

To get started, let's look at excerpts from examples of the C/V and XML orders that contain the verification information block.

```
[VerificationInformation]
VerifiedClaim: name
VerifiedClaim: address
VerificationResult: success
VerificationReference: ABC123/45GHT
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: idcard
VerificationMethod: auth
TrustFramework: de_denic
```

```
<verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0"
xsi:type="verification:verificationInformationType">
  <verification:verifiedClaims>
    <verification:claim>name</verification:claim>
    <verification:claim>address</verification:claim>
```

```
  </verification:verifiedClaims>
  <verification:verificationResult>success</verification:verificationResult>
  <verification:verificationReference>ABC123/45GHT</verification:verificationReference>
  <verification:verificationTimestamp>2023-11-11T15:36:21+02:00</verification:verificationTimestamp>
  <verification:verificationEvidence>idcard</verification:verificationEvidence>
  <verification:verificationMethod>auth</verification:verificationMethod>
  <verification:trustFramework>de_denic</verification:trustFramework>
</verification:verificationInformation>
```

Structure and rules for the verification information block for contacts of the types PERSON and ORG

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
[Veri-fic-ationIn-formation]	<veri-fication:verificationInformation xmlns:veri-fic-ation-="h-ttp://re-gistry.den-ic.de/verification/5.0" xsi:-	-	-	Once per veri-fic-ation recor-d	-	-	For K/V: header line that pre-cedes the veri-fic-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	type- ="ver- ific- ation:veri- ficationInformationType"> <verification:verifiedClaims> <veri- fication:verificationResult> <veri-						ation record

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fication:verificationReference> <veri- fic- ation:verificationTimestamp> <veri- fication:verificationEvidence> <veri- fication:verificationMethod> </veri-						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fic-ation:verificationInformation>						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
-	-	<veri-fic-ation:verifiedClaims> <verification:claim> </veri-fic-ation:veri-fiedClaims>	-	-	-	-	-

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerifiedClaim	-	-	<verification:claim> ... <verification:claim>	1 - 3	normalizedString / fixed length due to predefined valued	claim-rule	"claims" are data that has been verified **Special features for e-mail** The

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							com-plete veri-fication inform-ation block with all keywor-ds or XML ele-ments and val-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							must be spe-cified for an email, if the res-ult of the email check was neg-ative, in other words,

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							A sub-sequent update with the positive value ‘suc-cess’ must also be com-munic-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							ated, after a fail. If the value 'suc-cess' was present at the start of the check, the veri-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							fication inform-ation block for email **can** be repor-ted.

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationResult	-	<veri-fic-ation:veri-ficationResult> ... </veri-fic-ation:veri-ficationResult>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	res-ult-rule	Noti-fication of the veri-fication result

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationRe-ference	-	<veri-fic-ation:veri-ficationReference> ... /veri-fic-ation:veri-ficationReference>	-	1	nor-mal-izedStrin-g / engthis defined by mem-ber	ref-erenc-e-rule	The con-tent is free text that makes ref-erence to a request number or order num-ber, etc.

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationTimesta-mp	-	<veri-fic-ation:veri-fic-ationTimestamp> ... </veri-fic-ation:veri-	-	1	date-time	times-tamp-rule	Point in time, when the veri-fication was per-formed

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
		fic- ationTimestamp>					

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationEvid-ence	-	<veri-fic-ation:veri-ficationEvidence> ... </veri-fic-ation:veri-ficationEvidence>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	evid-ence-rule	Evid-ence that was checked within the scope of the veri-fication (e.g. for the value "idcar-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							d", which cor-respon-ds to identity card)

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationMethod	-	<veri-fic-ation:veri-ficationMethod> ... </veri-fic-ation:veri-ficationMethod>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	meth-od-rule	Method used to carry out the veri-fication (e.g. for the value "pvr", which stands for evid-encing via "video

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							iden-tific-ation")

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
TrustFrame-work	-	<veri-fic-ation:trustFrame-work> ... </veri-fic-ation:trustFrame-work>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	frame-work-rule	Frame-work used for the veri-fication At present, only the value "de_denic" exists, others may fol-low

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							later.

Rules for the value range

Rule Name of the Value Range	Value Range	Rules
claim-rule	email, name, address	No distinction is made between upper and lower case
result-rule	success, failed	No distinction is made between upper and lower case

Rule Name of the Value Range	Value Range	Rules
reference-rule	-	free text

Rule Name of the Value Range	Value Range	Rules
timestamp-rule	Year = 0000 - 9999 Month = 01 - 12 Tag = 01-28, 01-29, 01-30, 01-31 (based on month and year) Hour = 00 - 23 Minute = 00 - 59 Second = 00-58, 00-59, 00-60 (based on the Leap second)	You find a detailed description in the RFC3399: Date and Time on the Internet: Timestamps. The RFC uses the coordinated universal time as the basis for date-time definitions. DENIC expects the "date-time" format described in the RFC at this point. This format is composed of the complete date, consisting of year (y) with four characters, hyphen, month (m) with two characters, and day (d) with two characters. The date is followed by the time, which starts with the letter "T" to separate the two. The date is written in a 24-hour format, consisting of hour (h) with two characters, colon, minute (m) with two characters, colon, and second (s) with two characters. Finally, the time zone is added, introduced by a plus sign, followed by the time zone in hours (h) with two characters and the minute (m) with two characters. **Format** yyyy-mm-ddThh:mm:ss+mm:ss

Rule Name of the Value Range	Value Range	Rules
		Example 2024-04-02T14:58:30+02:00 This time is a “date-time” format for Central European Summer Time, which corresponds to the following local time: Date: 02.04.2024, Time: 12:58:30 and Universal Time Coordinated (UTC) with the date: 02.04.2024 and the time: 12:58:30.
evidence-rule	idcard, passport, population_register, residence_permit, proof_of_arrival, drivers_licence, company_register company_statement, bank_account, online_payment_account, utility_account, bank_statement, tax_statement, written_attestation, digital_attestation, postal_ver_transaction_log, email_ver_transaction_log, address_database	No distinction is made between upper and lower case
method-rule	auth, electronic_document, physical_document, vdig, bvr, pvr, data, reachability	-

Rule Name of the Value Range	Value Range	Rules
framework-rule	de_denic	At present, only the predefined text "de.denic" can be entered. In the future, other values may be possible, for instance when other registries operating in the European area offer verification procedures that are accepted by DENIC. By stating the other verification procedure, a member can inform that the claims were already checked and that the processes of DENIC should take this into consideration when processing the request.

Mapping table for method names to value names

Proof	Value in the Request
eID / eIDAS	auth
Electronic document record	electronic_document
Physical document record	physical_document

Proof	Value in the Request
PostIdent	vdig
Photo-Ident	bvr
Video-Ident	pvr
Transaction record	data
Reachability	reachability

Claims, verifications and methods that can be combined

- Multiple claims can occur per verification information.
- Only one verification and only one method can be specified per verification information block.

Proof	VerifiedClaim	VerificationEvidence	VerificationMethod (only one method per evidence allowed)
ID card	• name • address (if applicable)	idcard	• auth • vdig • electronic_document • physical_document • bvr • pvr
Passport	name	passport	• vdig • electronic_document • physical_document • bvr • pvr

Proof	VerifiedClaim	VerificationEvidence	VerificationMethod (only one method per evidence allowed)
Population register	• name • address	population_register	• electronic_document • physical_document • bvr • pvr
Residence permit	• name • address	residence_permit	• electronic_document • physical_document • bvr • pvr

Proof	VerifiedClaim	VerificationEvidence	VerificationMethod (only one method per evidence allowed)
Proof of arrival	name	proof_of_arrival	• electronic_document • physical_document • bvr • pvr
Driver's license	name	drivers_licence	• electronic_document • physical_document • bvr • pvr
Company register	• name • address	company_register	electronic_document

Proof	VerifiedClaim	VerificationEvidence	VerificationMethod (only one method per evidence allowed)
Founding documents	• name • address	company_statement	• electronic_document • physical_document
Federal Gazette	name	company_register	electronic_document
Record of payment via direct debit / transfer or credit card	name	bank_account	data
Record of payment via online payment	• name • address • email	online_payment_account	data
Utility bill	• name • address	utility_account	• electronic_document • physical_document

Proof	VerifiedClaim	VerificationEvidence	VerificationMethod (only one method per evidence allowed)
Bank statement	• name • address	bank_statement	• electronic_document • physical_document
Tax statement	• name • address	tax_statement	• electronic_document • physical_document
Written confirmation by an already verified natural person	• name • address • email	written_attestation	physical_document
Digital confirmation by an already verified natural person	• name • address • email	digital_attestation	electronic_document

Proof	VerifiedClaim	VerificationEvidence	VerificationMethod (only one method per evidence allowed)
Postal transaction log	• name • address	postal_ver_transaction_log	• vdig • reachability
E-mail transaction log	email	email_ver_transaction_log	reachability
Address Database	address	address_database	data

Copy and paste

For quickly inserting it into your source code, database, etc.

name, address, idcard, auth, vdig, electronic_document, physical_document, bvr, pvr

name, passport, vdig, electronic_document, physical_document, bvr, pvr

name, address, population_register, electronic_document, physical_document, bvr, pvr

name, address, residence_permit, electronic_document, physical_document, bvr, pvr

name, proof_of_arrival, electronic_document, physical_document, bvr, pvr

name, drivers_licence, electronic_document, physical_document, bvr, pvr

name, address, company_register, electronic_document

name, address, company_statement, electronic_document, physical_document

name, company_register, electronic_document

name, bank_account, data

name, address, email, online_payment_account, data

name, address, utility_account, electronic_document, physical_document

name, address, bank_statement, electronic_document, physical_document

name, address, tax_statement, electronic_document, physical_document

name, address, email, written_attestation, physical_document

name, address, email, digital_attestation, electronic_document

name, address, postal_ver_transaction_log, vdig, reachability

email, email_ver_transaction_log, reachability

address, address_database, data

More information

What is verification? This topic gives you some basics to get you started.

"Verification" on page 61

More about proofs, claims and methods and how to combine them.

Identity verification – methods, verification and claims

Sub-Title1

06 Requests to the .de Registration System

Requests to the .de Registration System 113
expireWarning and expire Message Type 113
Data Formats Data Types and Value Ranges 116
Description of Data Formats (Key/Value / XML) 117
Error Messages from the .de Registration System 119
Data Types and Value Ranges for Data Fields 125
Login and Logout 135
Login Request 137
Logout Request 141
Replies to LOGIN andLOGOUT Requests 143
Error Messages in Relation with LOGIN and LOGOUT Requests 145
Contact Requests 146
Contact Data Object 149
Information on the Phone Number 160
Explanation of the URI-Template 168
contactCREATE Request 170
contactUPDATE Request 210
Replies to contactCREATE and contactUPDATE Requests 251
contactCHECK Request 254
Replies to contactCHECK Requests 257
contactINFO Request 260
Replies to contactINFO Requests 297
Error Messages Related to Contact Requests 303
Requests Related to the Domain Inventory 309
Domain Statuses pendingCreate and serverHold 310
Domain Data Object 312
domainCREATE Request 324
domainUPDATE Request 327
domainCHHOLDER Request 331
domainRESTORE Request 335
domainDELETE Request 338
domainTRANSIT Request 342
Replies to Domain CREATE, UPDATE, CHHOLDER, RESTORE, DELETE, TRANSIT Requests 349
domainCHECK Request 351
domainINFO Request 356
Replies to domainCHECK and domainINFO Requests 386
Error Messages Related to Domain Requests 390
Requests Related to Public Contact Details of the Registrar 400

Public Registrar Contact Details Data Object 401
regaccINFO Request 412
Reply to regaccINFO Requests 416
Error Messages Related to RegAcc Requests 422
Provider Change Requests 422
Chprov Data Object 423
domainCREATE-AUTHINFO1 Request 425
domainCREATE-AUTHINFO2 Request 429
domainDELETE-AUTHINFO1 Request 434
domainCHPROV Request 438
Replies to CREATE-AUTHINFO1, CREATE-AUTHINFO2, CHPROV 445
Error Messages Related to Provider Change Requests 447
Requests Related to the Message Queue 455
QUEUE-READ Request 459
QUEUE-READ Responses 461
QUEUE-DELETE Request 461
QUEUE-DELETE Response 463
Error Messages Related to QUEUE-READ and QUEUE-DELETE Requests 464
authInfo2Delete Message Type 466
authInfo2Notify Message Type 468
authInfoExpire Message Type 470
chprovAuthInfo Message Type 473
contactDelete Message Type 475
contactVerificationRequired Message Type 477
domainDelete Message Type 485
domainStatusUpdate Message Type 505
expireWarning and expire Message Type 522

Requests to the .de Registration System

Overview

Introduction

Instructions to the .de registration system are submitted in form of requests (e.g. request for creation of a domain). In this section, you will find information about all the request types which are used in relation with the .de registration system.

Contents

This chapter comprises the following sections:

- Data Formats, Data Types and Value Ranges
- LOGIN / LOGOUT
- Contact Requests
- Requests Related to the Domain Inventory
- Requests related to the Public Contact Details of the Registrar
- Provider Change Requests
- Requests Related to the Message Queue

expireWarning and expire Message Type

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description
Result	result	1	enu-meration	result-erule	Result of pro-cessing success = order was car-ried out failed = order was not car-ried out exception = system mal-function, request was not carried out
StId	stid	1	token 30-255	stid-rule	A server trans-action ID (StId) assigned by the server. The StId is used to glob-ally and unam-biguously ref-erence the provider's request and the cor-responding response from DENIC.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgCnt	msgcnt	1	token 1 - 7	digit-rule	Without "MsgType", the total number of messages is displayed; with "MsgType", the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.
MsgTime	msgtime	1	dateTime 25	date-rule	The time of message creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description
MsgType	-	1	enu-meration	msgtype-erule	Two different message types: ExpireWarn-ing comes one week before the domain expires. Expire comes on the day the domain expired.
Domain	domain:hand-le	1	token 4 – 66	domain-rule	Domain name
Domain-Ace	domain:ace	0-1	token 4 - 66	domain-ace-rule	Domain name in ACE form (ASCII Compatible Encoding).
Expire	expire	0-1	dateTime 25	date-rule	Domain expir-ation date

Data Formats Data Types and Value Ranges

Overview

Introduction

In this section you will find information about the available data formats, data types and value ranges of data fields.

Contents

This section covers the following topics:

- Description of Data Formats (Key/Value / XML)
- Data Types and Value Ranges for Data Fields

Description of Data Formats (Key/Value / XML)

Request format: Key/Value

There is no defined sequence of keywords. You can choose freely.

Versioning of Key/Value Requests

At present, the valid version for requests in the key/value format is 4.0.

Validation in Case of Key/Value

The following rules apply:

- Keywords can be stated in case-insensitive form.
- A keyword must always be positioned at the beginning of a line.
- The keyword is always followed by a colon as assignment character (":"; 0x3A).
- There must be a gap between the assignment character and the assigned value comprised of at least one space (%x20) or one tabulator (%x09).
- Unknown keywords will be ignored.
- If optional keywords are omitted during an update the values previously set for them will be deleted.
- If an optional keyword is omitted and a default value exists, the default value will be set.

Request Format: XML

All XML processors must be able to recognize and process both UTF-8 and UTF-16 encodings. XML has the XML declaration encoding attribute, which enables it to identify and use other character encodings. If neither this attribute nor a BOM attribute is provided, UTF-8 encoding is set by default. This is the reason why the RRI server uses UTF -8 encoding and why it is recommend to RRI clients.

The XML format follows clearly defined schemata. The sequence of the XML elements must be strictly observed. Any deviation from the defined syntax will be rejected with an error message. Thus, unknown elements will also prompt an error. Elements in the XML format are case-sensitive! You must always enter a value for optional elements. If you want to delete an optional element, you must omit the complete line.

Version to be Used for XML

XML requests and the related replies can be validated against the version 4.0 schemata.

Validation in Case of XML

When the XML request format is used, requests may be validated via the following schemata:

- contact-4.0.xsd
- dnsentry-4.0.xsd
- domain-4.0.xsd
- global-4.0.xsd
- message-4.0.xsd
- monitoring-4.0.xsd
- regacc-4.0.xsd
- transaction-4.0.xsd

You need an XML-W3C-schema-1.0-compatible schema validator for this purpose. You will find the schemata on our member website (Schemata and BNFs).

Validation Errors in XML Requests

In connection with error code 83000000004 (e.g. in the table Error Messages Related to Domain Requests), which indicates that the validation of the XML request against the schema failed, XML parser information can be returned in several lines.

Each line is enclosed by the tag pair **<tr:argument>...</tr:agument>**.

The XML parser information is provided as a separate message of the software used by DENIC. It is not part of the RRI error messages defined by DENIC.

The wording of the XML parser information is determined by the software producer. Maintenance and care is not provided by DENIC. For this reason, the technical documents of DENIC do not include lists with or texts about those error codes.

The example below shall illustrate where in the XML response the XML parser information (in red) is to be expected if a validation error occurs.

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/3.0"
xmlns:tr="http://registry.denic.de/transaction/3.0">
<tr:transaction>
    <tr:stid>f099f01a-804d-11ec-b126-9f4bad0b1575</tr:stid>
    <tr:result>failed</tr:result>
    <tr:message level="error" code="83000000004">
      <tr:text>Validation of XML request against Schema
failed</tr:text>
      <tr:argument>cvc-elt.1.a: Cannot find the declaration of
element 'registry-request'.</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response>
```

Response Formats

RRI will always use the same format for its reply as was used for the request, i.e. either key/value or XML.

Error Messages from the .de Registration System

Potential Error Messages

Below you will find a list of the error messages you may receive in relation with any request, regardless of the request type.

Comments on the Table

- Error Code: Error coded according to Message Codes
- Error Message: Short text explanation in English for the error code
- Output format: "Error Code" and "Error Message" in K/V or XML format

Error Code	Error Message	Output Format	Brief Description
13000000011	Request was processed in test environment - not valid in real world	K/V, XML	This INFO message is the response to any request submitted in the test environment.
13000000022	Used quota	K/V, XML	Once a defined threshold value is reached, this INFO message informs you about the percentage of your quota you have already used up.
33000000022	Used quota	K/V, XML	Once a defined threshold value is reached, this WARNING message informs you about the percentage of your quota you have already used up.

Error Code	Error Message	Output Format	Brief Description
53000000008	Daily quota exceeded	K/V, XML	This error message is prompted when the available quota is exhausted.
53000000012	Concurrent write access to the same business object is prohibited [dynamic error text]	K/V, XML	This error is prompted with any writing request, if simultaneous writing access is made to one of the objects included in the request.
53000020002	Unauthorized request	K/V, XML	This error message informs you that you are not authorized to submit this type of request. Please contact Business Services.
60009500901	The PGP key does not comply with the requirements: [dynamic error text]	K/V, XML	This error message informs you that your key does not comply with DENIC's key policy.
63000000006	The request has an invalid key/-value pair format	K/V	You have indicated an invalid key/value pair.

Error Code	Error Message	Output Format	Brief Description
63000000007	Request shows semantics error [dynamic error text]	K/V	Incorrect semantics, the error occurs in the context of non-writing order forms, e.g. Domain-INFO with the specification of an unexpected CTID.
63000000012	Unknown Error	K/V, XML	An unknown error has occurred in the RRI. Send the error message and a description of the situation in which the error occurred to DBS via chat or email.
63000040110	Keyword “Action” cannot be recognized	K/V	The keyword "Action" is missing.
63000040410	Keyword “Version” cannot be recognized	K/V	The keyword "Version" is missing.
63000060003	An indicated Contact is invalid [repetition of the invalid handle]	K/V, XML	A Contact is invalid.

Error Code	Error Message	Output Format	Brief Description
63000060008	The value contains one or more invalid characters [dynamic error text]	K/V	You have stated one or several invalid characters for this value.
63000060015	The value is too long [dynamic error text]	K/V	The value is too long.
63000060102	Inadmissible value for keyword “Action”	K/V	The value entered for the keyword "Action" is not permitted.
63000060402	Inadmissible value for keyword “Version”	K/V	The value entered for the keyword "Version" is not permitted.
63000060504	“CTID” has been used before [dynamic error text]	K/V, XML	The value entered for "CTID" has been used before.
63000060508	Inadmissible value for keyword "Ctid"	K/V	The value entered for "CTID" is invalid.
63300080010	Required domain data is missing	K/V	Required domain data is missing.
83000000000	Request message too long [dynamic error text]	K/V	Request message is too long.

Error Code	Error Message	Output Format	Brief Description
83000000002	The encoding specified is not supported	K/V	The encoding you have stated is not supported.
83000000004	Validation of XML request against Schema failed	K/V, XML	The syntax of the XML request is incorrect, validation against the schema fails.

Data Types and Value Ranges for Data Fields

Introduction

This section gives conceptual definitions for all data types and value ranges of the data fields used in requests and replies. The descriptions make reference to the requests and replies used in the .de registration system.

The requests and replies of the registration system may include Unicode characters, i.e. also non-ASCII characters, such as ĕ und þ.

Data Types

Below you will find information about the data types which are used:

Data Types	Comment
normalizedString	A normalizedString represents white space normalized strings. The value range of normalizedString is the set of strings that do not contain the carriage return (%xD), line feed (%xA) or tab (%x9) characters. For detailed information about the data type go to normalizedString.
token	A token represents tokenized strings. The value range of token is the set of strings that do not contain the carriage return (%xD), line feed (%xA) or tab (%x9) characters, that have no leading or trailing spaces (%x20) and that have no internal sequences of two or more spaces. For detailed information about the data type token go to token .

Data Types	Comment
enumeration	Enumeration constrains the value range to a specified set of values. For detailed information about enumeration go to enumeration .
date-time	The data type datetime is used to state a date and a time and a specific time zone. The syntax follows the rules stipulated in the standard ISO 8601: • Dates are written: YYYY-MM-DD. • Then follows T (for time) as a separator between date and time. • Then follows the time, which is written: hh:mm:ss. • Then follows recommended information, the difference to the coordinated universal time (UTC). • It is written: +hh:mm Example: 2010-04-07T13:16:00+01:00
unsignedShort	The data type unsignedShort designates an integer value in the value range 0 through 65535. For detailed information about the data type unsignedShort go to unsingedShort .
boolean	boolean may have either of two values - "true" and "false". For detailed information about the data type boolean go to boolean .

Rules for Value Ranges

In this section, you will find detailed information about the syntax specifications for the value ranges of individual keywords. (Augmented BNF for Syntax Specifications: ABNF).

Type	Value Ranges	Definition
ace-val	%xx41-%x5A / %x61-%x7A / %x2D	A-Z / a-z / hyphen
alpha-val	%x41-%x5A / %x61-%x7A	A-Z / a-z
authinfo-val	%x41-%x48 %x4A-%x4E %x50-%x5A %x61-%x6B %x6D-%x6E %x70-%x7A / %x32-%x39 / %x2B / %x2D / %x2F / %x2A	A-H J-N P-Z a-k m-n p-z / 2-9 / plus sign / hyphen / slash / asterisk
digit-val	%x30-%x39	0-9
dnskeyval	%x41-%x5A / %x61-%7A / %x30-39 / %x2D / %x2E / %x2F / %x2B	A-Z / a-z / 0-9 / hyphen / dot / slash / plus sign

Type	Value Ranges	Definition
idn-val	%xE1 / %xE0 / %x103 / %xE2 / %xE5 / %xE4 / %xE3 / %x105 / %x101 / %xE6 / %x107 / %x109 / %x10D / %x10B / %xE7 / %x10F / %x111 / %xE9 / %xE8 / %x115 / %xEA / %x11B / %xEB / %x117 / %x119 / %x113 / %x11F / %x11D / %x121 / %x123 / %x125 / %x127 / %xED / %xEC / %x12D / %xEE / %xEF / %x129 / %x12F / %x12B / %x131 / %x135 / %x137 / %x13A / %x13E / %x13C / %x142 / %x144 / %x148 / %xF1 / %x146 / %x14B / %xF3 / %xF2 / %x14F / %xF4 / %xF6 / %x151 / %xF5 / %xF8 / %x14D / %x153 / %x138 / %x155 / %x159 / %x157 / %x15B / %x15D / %x161 / %x15F / %x165 / %x163 / %x167 / %xFA / %xF9 / %x16D / %xFB / %x16F / %x171 / %x169 / %x173 / %x16B / %x175 / %xFD / %x177 / %xFF / %x17A /17E / %x17C / %xF0 / %xFE / %xDF	áàăâåäãąāæćĉčċçďđéèĕêěëėęēğĝġģĥħíìĭîĩïįīıĵķĺľļłńňñņŋóòŏôöőõøō oeĸŕřŗśŝšşťţŧúùŭûůüűũųūŵýŷÿźžżðþß
ip-val	%x30-%x39 / %x2E / %x3A	ASCII: 0-9 / dot / colon
name-val	%x41-%x5A / %x61-%x7A / %x30-%x39 / %x20 / %x26 / %x2D / %x2E / %xA0-%xFF	ASCII: A-Z / a-z / 0-9 / space / ampersand (@) / hyphen / dot / visible characters from Latin-1
nserver-val	%xx41-x5A / %x61-%x7A / %x30-%x39 / %x2D / %x2E / %x3A / %x20	ASCII: A-Z / a-z / 0-9 / hyphen / dot / colon/ space

Applicable Rules
address-rule = 1*255(name-val)
authinfohash-rule = 64*64(ace-val / digit-val)
authinfo-erule = 1*1("true")
authinfo-rule = 1*1(authinfo-val)
boolean-erule = 1*1("true" / "false")
check-erule = 1*1("CHECK")
chholder-erule = 1*1("CHHOLDER")
chprov-erule = 1*1("CHPROV")
city-rule = 1*80(name-val)
contact-rule = 9*32"DENIC-" (digit-val / ace-val / %x2E)
contactstatus-erule = 1*1("free" / "exists" / "invalid")

Applicable Rules
country-erule = 1*2(ALPHA) **Hint** The valid list of ISO country codes (country-erule) can be found here: ISO 3166 Country Codes
creauthinfo1-erule = 1*1("CREATE-AUTHINFO1")
creauthinfo2-erule = 1*1("CREATE-AUTHINFO2")
create-erule = 1*1("CREATE")

Applicable Rules
date-rule = 25*25(digit-val / %x3A / %x2D / %x2B / %x54) • 0-9 (%x30-%x39) • colon (%x3A) • hyphen (%x2D) • plus sign (%x2B) • character "T" (%x54)
delauthinfo1-erule = 1*1("DELETE-AUTHINFO1")
delete-erule = 1*1("DELETE")
digit-rule = 8*9(digit-val)
domain-rule = 4*66(idn-val / ace-val) %x2E "de" • The dot (%x2E) acts as a seperator between the domain and the TLD.

Applicable Rules
domainace-rule = “xn--" 8*66((ace-val / digit-val) ace-val) %x2E "de" • Domain-Ace begins with xn-- (%x78 %x6E %x2D %x2D) • The dot (%x2E) acts as a seperator between the domain and the TLD.
dnskey-rule = 30*999(dnskeyval)
email-rule = 3*255(ace-val / digit-val / %x2E / %x40) • A-Z • a-z • 0-9 • hyphen (%x2D) • dot (%x2E) • At sign (%x40)
ip-rule = 7*39(ip-val)
info-erule = 1*1("INFO")

Applicable Rules
login-erule = 1*1("LOGIN")
logout-erule = 1*1("LOGOUT")
message-rule = 1*255(ace-val / digit-val / idn-val / %x20)
msgid-rule = 30*255(ALPHA / digit-val / %x2D)
mx-rule = 4*252(nserver-val)
mxpref-rule = 1*3(digit-val)
name-rule = 1*255(ace-val / idn-val / digit-val / name-val)
nsentry-rule = 4*252(nserver-val)
nserver-rule = 1*254(nserver-val)
organisation-rule = name-rule

Applicable Rules
phone-rule = 1*255(%x2B) (digit-val / %x2E / %x2D / %x78 / %x20 / %x2F) • plus sign (%x2B) • 0-9 • dot (%x2E) • hyphen (%x2D) • character "x" (%x78)# • space (%x20) • slash (%x2F)
postalcode-rule = 1*20(alpha-val / digit-val / %x2D / %x20) • hyphen (%x2D) • space (%x20)
queueread-erule = 1*1("QUEUE-READ")
queuerdelete-erule = 1*1("QUEUE-DELETE")

Applicable Rules
regaccid-rule = ”DENIC-" 7*13(digit-val)
restore-erule = 1*1("RESTORE”)
result-erule =1*1("success" / "failed" / „exception”)
role-erule = 1*1("PERSON" / "ORG" / “REQUEST”)
status-erule = 1*1("connect" / "failed" / "invalid" / "free" / "pendingCreate" / "serverHold")
stid-rule = 30*255(ace-val / digit-val)
transit-erule = 1*1("TRANSIT")
update-erule = 1*1("UPDATE")
version-erule = 1*1(”3.0”)

Login and Logout

Overview

Introduction

In this section, you will find information about LOGIN / LOGOUT.

Contents

This section covers the following topics:

- LOGIN
- LOGOUT
- Replies to LOGIN, LOGOUT Requests.
- Error Messages in Relation with LOGIN/LOGOUT

Login Request

Function

The login establishes a session with RRI. The login is transmitted to RRI in form of a request.

Features

If there are 10 consecutive incorrect logins, the user will be blocked for one hour and all further login attempts will be rejected. As long as the blocking is active, no logins are permitted, even if the data is valid.

- Login requests are not part of the quotas.
- The password should:
    - have at least 12 characters,
    - have a sufficiently secure entropy (> 70 bit),
    - be changed immediately in the event of loss or compromise,
    - always be created anew and must not correspond to a previously used password.

Request Parameters

An request consists of the keywords or elements listed below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	login	1	enumeration	login-erule	request type
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/value format.
User	user	1	token 9 – 32	contact-rule	An RRI login must be used as value for the user. The RRI login is stored in the RegAcc profile in the "RRI Login" line. The RegAcc profile is maintained via the RAI (see documentation "RAI - Registrar Administration Interface", How to Edit the RegAcc Profile).
Password	password	1	token 1 - 255	Any visible Unicode char-acter (according to Unicode ver-sion 3.1	This is the user password in clear text, no value range is defined.

Examples

- xml
- k/v

format: xml

request: logout

```
1 | <!-- request -->
2 |
3 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4 |
5 | <registry-request xmlns="http://registry.denic.de/global/5.0">
6 |  <logout/>
7 | </registry-request>
```

format: xml

response: logout

```
1 | <?xml version="1.0" encoding="UTF-8"?>
```

```
2
3  <registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="h-
   ttp://registry.denic.de/transaction/5.0">
4    <tr:transaction>
5      <tr:stid>078db1bf-9f0a-11df-a9e2-9519b5688882</tr:stid>
6      <tr:result>success</tr:result>
7    </tr:transaction>
8  </registry-response>
```

format: k/v

request: logout

```
1  Version: 5.0
2  Action: logout
```

format: k/v

response: logout

```
1  Result: success
2  STID: 0b97f9b7-9f0b-11df-a9e2-9519b5688882
```

Logout Request

Function

The logout terminates a session established with RRI. The logout is transmitted to RRI in form of a request.

Features

If the connection is cut without logging out, the session with the RRI server will remain active for a certain period. Thus, you may be unable to log in to the RRI-Server for another session because 7 concurrent sessions may already be running. Logout requests do not reduce the quota.

Request Parameters

A LOGOUT request is composed of the keywords and/or elements listed below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	logout	1	enumeration	logout-erule	request type
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/-value format.

Examples

- xml
- k/v

format: xml

request: logout

```
<!-- request -->

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0">
 <logout/>
</registry-request>
```

format: xml

response: logout

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>078db1bf-9f0a-11df-a9e2-9519b5688882</tr:stid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: k/v

request: logout

```
Version: 5.0
Action: logout
```

format: k/v

response: logout

```
Result: success
STID: 0b97f9b7-9f0b-11df-a9e2-9519b5688882
```

Replies to LOGIN andLOGOUT Requests

Beschreibung der Datenfelder

Die Antwort des .de-Registrierungssystems auf ein LOGIN / LOGOUT-Kommando beinhaltet folgende Datenfelder:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Valu Range	Description
Result	result	1	enumeration	result-erule	This is the processing result. success = request was executed. failed = request was not executed. exception = system failure. Request was not executed.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Valu Range	Description
StId	stid	1	token 30 - 255	stid-rule	This is a Server Transaction ID (StId) which is assigned by the server. The StId ensures that the request from the provider and DENIC's reply are referenced with a globally unique ID. The StId is output only in reply to a LOGIN command. **Note on LOGOUT:** The STID is only sent if the logout is successful. The message with the code 13000000011 in the test environment is also only displayed in the event of a successful logout.

Error Messages in Relation with LOGIN and LOGOUT Requests

Potential Error Messages

Below you will find a list of the error messages you may receive in relation with session management and LOGIN and LOGOUT..

Comments on the Table

- Error Code: Code created after message codes
- Error message: short error message that is output with the error code
- Output format: The message can be output in KV (Key/Value) and/or XML formats.
- LOGIN/LOGUT: The message occurs with LI (login) or LO (logout).
- Description: Explanation of the message

Error Code	Error Message	Output Format	LOGIN (LI) LOGOUT (LO)	Description
63000040110	Keyword "Action" cannot be recognized	KV	LI/LO	The keyword "Action" could not be recognized.
63100020005	Too many failed logins, login is temporarily blocked	KV / XML	LI	In order to protect your access against abuse, it is blocked for a defined period after a defined number of failed login attempts.
63100040202	Keyword "User" must appear once and just once	KV	LI	You entered the keyword "User:" several times.

Error Code	Error Message	Output Format	LOGIN (LI) LOGOUT (LO)	Description
63100040302	Keyword "Password" must appear once and just once	KV	LI	You forgot to enter the keyword "Password:" or you entered it several times.
73100000005	The maximum number of permitted login is reached	KV / XML	LI	To protect the system against excessive use (i.e. abuse), only a defined limited number of concurrent sessions is permitted. If this maximum is reached, any further attempt to set up a session (login) will be rejected.
83000000004	Validation of XML request against Schema failed	XML	LI/LO	The syntax of the XML request is incorrect, validation against the schema fails.
83000000010	Login failed	KV / XML	LI/LO	You entered an invalid user or password.

Contact Requests

Overview

Introduction

The Contacts are used to administer personal and organisational data as well as contact information (to submit general and technical requests or to submit enquiries and information about possible unlawful or improper use of a domain).

Before you can register a domain you must have entered data about the domain holder (Holder) and optional data about the contact information (General Request and Abuse Contact).

This information is stored in so-called Contacts. Each Contact has a unique handle ID, which in turn is used to make reference to the Contact in domain requests.

This section describes the data object "Contact" and the request types you can use in relation with it.

Contents

This section covers the following topics:

- Contact Data Object
- Contact CREATE Request
- Contact UPDATE Request
- Replies to Contact CREATE and UPDATE
- Contact CHECK Request
- Contact INFO Request
- Replies to Contact INFO Requests
- Error Messages Related to Contact Requests

Important Information

- Before you can register a domain you have to create the necessary Contacts. If you register any additional domains or make further domain requests you can fall back on existing Contacts.

- If you want to collect data of future domain holders such as name, address or e-mail address, you must inform the future domain holders of your activities in advance. You must state the purpose, scope and the intended use of the data. In addition to that you must inform your client that DENIC will also store client data when a domain is registered. You will provide this information by submitting the DENIC Domain Terms and Conditions to the client for information. You may also include a corresponding note in your own terms and conditions or in your order form to draw the client's attention to this fact.
- The requests contact CREATE and contact UPDATE must contain an e-mail address for the contact objects PERSON and ORG.
- Once you have created a Contact you cannot delete or deactivate it any more. It will always persist.

Contact Data Object

Contents of the Data Object

The contact data object contains all the relevant data about the

PERSON and ORG

with the contact data:

- unique handle,
- domain owners,
- address,
- email,
- telephone number,
- verification information,

REQUEST

with the contact details (for General Request or Abuse Contact):

- unique handle,
- URI template.

Processing Options

You can use the request forms Contact CREATE and Contact UPDATE to establish a new Contact or to change an existing one.

Structure and rules for contacts of the types PERSON and ORG

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<MemberID >	• A Contact handle must always start with the RegAccId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	These Contact types are available: • PERSON = natural person, • ORG = a legal person (company, association, grouping of holders, organization etc., such as "DENIC eG")	• The type of contact can be changed later. • For Holder you may only use the types PERSON or ORG.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Name	contact:name	1	nor-malizedString 1 - 255	name-rule	This is the Contact's name.	• Once set, you cannot change the keyword "name" any more.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Organisation	con-tact:organisation	0 - *	nor-malizedString 1 - 255	organisation-rule	This is the organ-isation which is represented by the Contact.	• You may only use the types PERSON. • Cannot be changed, if the domain bears a DISPUTE entry. • You must always state the domain holder in the "Name:" field. The "Organ-ization" value

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						has no right to the domain and is not entitled to apply for an AuthInfo or a provider change or similar.
Address	contact:address	1 - *	nor-malizedString 1 - 255	address-rule	Street and house number of Contact	The keyword cannot be changed if the Contact is ref-erenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
PostalCode	contact:postalCode	1	token 1 - 20	postalcode-rule	Postal code and address of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.
City	contact:city	1	normalizedString 1 - 80	city-rule	Place of residence of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
CountryCode	con-tact:countryCode	1	Enumeration 2	country-erule	Country code of the country in which the Contact's place of residence is loc-ated	• You must use the country code of the ISO-3166-1 alpha-2 list ISO-3166-1 alpha-2 Liste. • The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						entry.
Email	contact:email	1 - *	nor-malizedString 3 - 255	email-rule (see RFC5322 - Internet Message Format)	E-mail address of Contact	„email" always is a mandatory field and must be specified at least once for the types PERSON and ORG in Contact CREATE and Contact UPDATE requests

Contact Data Object: Syntax and Processing Rules for Contacts of the REQUEST Type

The table below informs about the contacts of the REQUEST type.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<Mem-berID >	A Contact handle must always start with the RegAc-cId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	REQUEST = an e-mail address or a URL in the URI-Template format.	For General Request and Abuse Contact you may only use the type REQUEST.
URI-Tem-plate	contact:uri-tem-plate	1	normalizedString 8 - 1024	Syntax as defined in RFC 6570 - URI Tem-plate	The variables Ula-bel and Alabel in the URI-Template are replaced with the domain at a domain query (web-whois).	In case of a CREATE request, the content of the URI-Template is converted into a URL or e-mail as a test (For Ala-bel and Ulabel, a sample domain is used.) to check the syntax on cor-rectness.

Information on the Phone Number

Specification of a Dummy Phone Number

Caution!

- Due to the legal situation, members can choose whether to provide a valid phone number or a dummy phone number for a contact.
- The format of the dummy phone number consists of the country code and the phone number, which consists only of zeros (0).
- The phone numbers or dummy phone numbers must be specified in the K/V or XML order.
- Specifying a dummy phone number is only possible on a temporary basis.
- In the future, DENIC will only accept valid phone numbers, which will be checked using the completeness check. All requests sent with a dummy phone number will be rejected by DENIC.
- DENIC will announce the date on which only valid phone numbers will be accepted via the usual information channels.

- The following rules apply to the format of a dummy phone number.

- Plus (“+”)
- Country code, consisting of 1 to 3 digits
- Period (“.”)
- Telephone number consisting of 4 to 14 zero digits (“0”)
- No optional information

Examples

Dummy expression for phone in K/V

2 examples

```
Phone: +49.0000
Phone: +43.000000
```

Dummy expression for phone in XML

2 examples

```
<contact:phone>+49.0000</contact:phone>
<contact:phone>+43.000000</contact:phone>
```

Request Parameters

K/V Keyword	XML Namespace and Element	Occurrence, min - max	Type / Length	Range / Value	Description	Policy
Phone	<contact:phone>...</contact:phone>	1	normalizedString	• Plus sign (%x2-B) • 0 - 9 (%x3-0-39) • Dot (%x2E) • x (%x7-8)	Phone number of the Contact	The EPP format is used for telephone numbers (Extensible Provisioning Protocol (EPP) Contact Mapping, Section 2.5) • Plus sign(1 character) • Country code (1 to 3 digits) • Dot (1 character) • Phone number (4 to 14 digits)

K/V Key-word	XML Namespace and Ele-ment	Occur-rence, min - max	Type / Length	Range / Value	Descrip-tion	Policy

						• **Optional data**
						  • Char-acter "x", fol-lowed by the
						  • tele-phone exten-sion

Examples and exact Value Range as a RegEx

```
# Examples
+49.69272350
+49.6927235123
+49.69123456x156

# Regular expression (also see https://www.oreilly.com/library/view/regular-expressions-cookbook/9781449327453/ch04s03.html)
^\+[0-9]{1,3}\.[0-9]{4,14}(?:x.+)?
```

- Telephone numbers that are not dummy values are displayed in the Contact INFO request.
- For Contact CREATE and Contact UPDATE requests, a phone number or dummy phone number is mandatory.
- Domain requests that result in a change of a domain owner handle will fail if a contact without a phone number is used. For multi-holder domains, this requirement applies to all domain owners.

Error Message K/V and XML

```
ERROR: 63200042402 Keyword "Phone" must appear once and just once
```

```
<tr:message level="error" code="63200042402">
    <tr:text>Keyword "Phone" must appear once and just once</tr:text>
</tr:message>
```

Explanation of the URI-Template

Short description of the URI-Template

URI-Templates are only used for REQUEST contacts (General Request / Abuse Contact). A URI-Template consists of a URI and optional variables.

In the case of a domain query (web-whois), the variables with the queried domain are replaced in the URI (URL or e-mail address).

URI-Template examples

Example with a URL:

A REQUEST contact contains the following URL with two variables for a URI-Template:

```
https://denic.de/contact/form?value1={Alabel}&value2={Ulabel}
```

The variables in the URL are enclosed in curly brackets and are DENIC's predefined variables for Alabel and Ulabel.

For a domain query (web-whois) with the example domain "beispiel-fünf.de", the domain is assigned to the variables as an ACE string and UTF-8 string:

- `Alabel := „xn--beispiel-fnf-mlb.de“`
- `Ulabel := „beispiel-fünf.de“`

In the URL, the variables are replaced with the strings:

```
https://denic.de/contact/form?value1=xn--beispiel-fnf-mlb.de&-
value2=beispiel-fünf.de
```

Example with an e-mail address:

For a REQUEST contact, with an e-mail address and a variable in the URI-Template, the variable is replaced in the case of a domain query (web-whois) with the example domain "beispiel-fünf.de".

Content URI-Template:

```
mailto:info@denic.de?subject=domain:{Ulabel}
```

Content variable:

`Ulabel := „beispiel-fünf.de“`

Ulabel is DENIC‘s predefined variable for domains represented as UTF-8 strings.

The variable is replaced with the string:

```
mailto:info@denic.de?subject=domain:beispiel-fünf.de
```

DENIC‘s rules for URI-Templates

- The URI-Template can be up to 1024 characters long.
- The URI schemas http, https and mailto are supported.
- DENIC has predefined two variables for URI templates:
  - Alabel: Domain is displayed as an ACE string.
  - Ulabel: Domain is represented as UTF-8 string.
- The variables are case-sensitive.
- The variables are optional.
- The use of other variables leads to an error.
- Both variables can be used multiple times in the URI-Template.
- The variables are replaced with the domain only for a domain query (web-whois).
- For an RRI INFO query, the URI-Template is displayed with the variables.

> **Notice**
>
> DENIC only supports level 1 as described in the RFC 6570 - URI Template.

contactCREATE Request

Function

This request is used for creating a new Contact.

Requirements

The Contact must not yet exist.

Verification

- Verification information is created for a contact of the PERSON or ORG type.
- The provision of verification information is optional.
- The provision of a telephone number is mandatory.

Request Parameters

A Contact CREATE request is composed of the "Contact" data object fields and some other parameters, which are described below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	contact:create	1	enumeration	create-erule	request type
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/value format.
CtId	ctid	0 - 1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

The Contact handle stated in the request already exists, see error messages related to Contact requests.

Data object for PERSON or ORG

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<MemberID >	• A Contact handle must always start with the RegAccId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	These Contact types are available: • PERSON = natural person, • ORG = a legal person (company, association, grouping of holders, organization etc., such as "DENIC eG")	• The type of contact can be changed later. • For Holder you may only use the types PERSON or ORG.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Name	contact:name	1	nor-malizedString 1 - 255	name-rule	This is the Contact's name.	• Once set, you cannot change the keyword "name" any more.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Organisation	con-tact:organisation	0 - *	nor-malizedString 1 - 255	organisation-rule	This is the organ-isation which is represented by the Contact.	• You may only use the types PERSON. • Cannot be changed, if the domain bears a DISPUTE entry. • You must always state the domain holder in the "Name:" field. The "Organ-ization" value

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						has no right to the domain and is not entitled to apply for an AuthInfo or a provider change or similar.
Address	contact:address	1 - *	nor-malizedString 1 - 255	address-rule	Street and house number of Contact	The keyword cannot be changed if the Contact is ref-erenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
PostalCode	contact:postalCode	1	token 1 - 20	postalcode-rule	Postal code and address of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.
City	contact:city	1	normalizedString 1 - 80	city-rule	Place of residence of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
CountryCode	con-tact:countryCode	1	Enumeration 2	country-erule	Country code of the country in which the Contact's place of residence is loc-ated	• You must use the country code of the ISO-3166-1 alpha-2 list ISO-3166-1 alpha-2 Liste. • The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						entry.
Email	contact:email	1 - *	nor-malizedString 3 - 255	email-rule (see RFC5322 - Internet Message Format)	E-mail address of Contact	„email" always is a mandatory field and must be specified at least once for the types PERSON and ORG in Contact CREATE and Contact UPDATE requests

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
[VerificationInformation]	<verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:-	-	-	Once per verification record	-	-	For K/V: header line that precedes the verific-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	type- ="ver- ific- ation:veri- ficationInformationType"> <verification:verifiedClaims> <veri- fication:verificationResult> <veri-						ation record

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fication:verificationReference> <veri- fic- ation:verificationTimestamp> <veri- fication:verificationEvidence> <veri- fication:verificationMethod> </veri-						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fic-ation:verificationInformation>						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
-	-	<veri- fic- ation:verifiedClaims> <verification:claim> </veri- fic- ation:veri- fiedClaims>	-	-	-	-	-

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerifiedClaim	-	-	<verification:claim> ... <verification:claim>	1 - 3	normalizedString / fixed length due to predefined valued	claim-rule	"claims" are data that has been verified **Special features for e-mail** The

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
							complete verification information block with all keywords or XML elements and val-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							must be spe-cified for an email, if the res-ult of the email check was neg-ative, in other words,

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							A sub-sequent update with the positive value ‘suc-cess’ must also be com-munic-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							ated, after a fail. If the value 'suc-cess' was present at the start of the check, the veri-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							fication inform-ation block for email **can** be repor-ted.

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationResult	-	<veri-fic-ation:veri-ficationResult> ... </veri-fic-ation:veri-ficationResult>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	res-ult-rule	Noti-fication of the veri-fication result

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationReference	-	<verification:verificationReference> ... /verification:verificationReference>	-	1	normalizedString / ength is defined by member	reference-rule	The content is free text that makes reference to a request number or order number, etc.

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationTimestamp	-	<verification:verificationTimestamp> ... </verification:verification	-	1	dateTime	timestamp-rule	Point in time, when the verification was performed

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
		fic- ationTimestamp>					

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationEvid-ence	-	<veri-fic-ation:veri-ficationEvidence> ... </veri-fic-ation:veri-ficationEvidence>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	evid-ence-rule	Evid-ence that was checked within the scope of the veri-fication (e.g. for the value "idcar-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							d", which cor-respon-ds to identity card)

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationMethod	-	<veri-fic-ation:veri-ficationMethod> ... </veri-fic-ation:veri-ficationMethod>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	meth-od-rule	Method used to carry out the veri-fication (e.g. for the value "pvr", which stands for evid-encing via "video

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							iden-tific-ation")

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
TrustFramework	-	<verification:trustFramework> ... </verification:trustFramework>	-	1	normalizedString / fixed length due to predefined valued	framework-rule	Framework used for the verification At present, only the value "de_denic" exists, others may follow

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							later.

Data object for REQUEST

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<Mem-berID >	A Contact handle must always start with the RegAc-cId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	REQUEST = an e-mail address or a URL in the URI-Template format.	For General Request and Abuse Contact you may only use the type REQUEST.
URI-Tem-plate	contact:uri-tem-plate	1	normalizedString 8 - 1024	Syntax as defined in RFC 6570 - URI Tem-plate	The variables Ula-bel and Alabel in the URI-Template are replaced with the domain at a domain query (web-whois).	In case of a CREATE request, the content of the URI-Template is converted into a URL or e-mail as a test (For Ala-bel and Ulabel, a sample domain is used.) to check the syntax on cor-rectness.

Examples

- xml
- k/v

format: xml

request: contactCREATE

type: PERSON

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:verification="http://registry.denic.de/verification/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <contact:create>
    <contact:handle>DENIC-1000022-EXAMPLE-XML-PERSON</contact:handle>
    <contact:type>PERSON</contact:type>
    <contact:name>John Doe</contact:name>
    <contact:organisation>DENIC eG</contact:organisation>
    <contact:postal>
      <contact:address>Theodor-Stern-Kai 1</contact:address>
      <contact:postalCode>60596</contact:postalCode>
      <contact:city>Frankfurt am Main</contact:city>
      <contact:countryCode>DE</contact:countryCode>
    </contact:postal>
```

```
15      <contact:email>john.doe@denic.de</contact:email>
16      <contact:phone>+49.69272350</contact:phone>
17      <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0"
    xsi:type="verification:verificationInformationType">
18          <verification:verifiedClaims>
19           <verification:claim>name</verification:claim>
20           <verification:claim>address</verification:claim>
21         </verification:verifiedClaims>
22         <verification:verificationResult>success</verification:verificationResult>
23         <verification:verificationReference>ABC123/45GHT</verification:verificationReference>
24         <verification:verificationTimestamp>2023-11-11T15:36:21+02:00</verification:verificationTimestamp>
25         <verification:verificationEvidence>idcard</verification:verificationEvidence>
26         <verification:verificationMethod>auth</verification:verificationMethod>
27         <verification:trustFramework>de_denic</verification:trustFramework>
28      </verification:verificationInformation>
29    </contact:create>
30    <ctid>xml-74ba5119</ctid>
31  </registry-request>
```

format: xml

response: contactCREATE

type: PERSON

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>f4dc86ae-8899-4768-8749-e149c85f9146</tr:stid>
    <tr:ctid>xml-74ba5119</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: xml

request: contactCREATE

type: REQUEST

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```
<registry-request xmlns="https://registry.denic.de/global/5.0" xmlns:contact="h-
ttps://registry.denic.de/contact/5.0">
<contact:create>
<contact:handle>DENIC-99995-GENREQ</contact:handle>
<contact:type>REQUEST</contact:type>
<contact:uri-template>https://denic.de/contact/form{?Alabel,Ulabel}</contact:uri-template>
</contact:create>
<ctid>cba-987654321</ctid>
</registry-request>
```

format: xml

response: contactCREATE

type: REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="h-
ttp://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>bb135322-958c-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
```

```
</registry-response>
```

format: k/v

request: contactCREATE

type: PERSON

```
Version: 5.0
Action: CREATE
Handle: DENIC-1000022-EXAMPLE-PERSON
Type: PERSON
Organisation: DENIC eG
Name: Max Mustermann
Address: Business Services
Address: Theodor-Stern-Kai 1
Address: in Frankfurt
Address: south side
Address: riverside
PostalCode: 60596
City: Frankfurt am Main
CountryCode: DE
eMail: email-1@denic.de
eMail: email-2@denic.de
eMail: email-3@denic.de
eMail: email-4@denic.de
eMail: email-5@denic.de
```

```
eMail: email-6@denic.de
Phone: +49.6927235x290
CTID: kv-7bf04fa8

[VerificationInformation]
VerifiedClaim: name
VerifiedClaim: address
VerificationResult: success
VerificationReference: ABC123/45GHT
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: idcard
VerificationMethod: auth
TrustFramework: de_denic

[VerificationInformation]
VerifiedClaim: email
VerificationResult: failed
VerificationReference: 354546TZQ
VerificationTimestamp: 2023-10-04T12:22:19+02:00
VerificationEvidence: email_ver_transaction_log
VerificationMethod: reachability
TrustFramework: de_denic
```

format: k/v

response: contactCREATE

type: PERSON

```
RESULT: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d
CTID: kv-7bf04fa8
```

format: k/v

request: contactCREATE

type: REQUEST

```
Version: 4.0
Action: create
Type: REQUEST
Handle: DENIC-99995-GENREQ
Uri-template: https://denic.de/contact/form{?Alabel,Ulabel}
```

format: k/v

response: contactINFO

type: REQUEST

```
Result: success
STID: 4cfcf3ff-958c-11df-a9e2-9519b5688882
```

contactUPDATE Request

Function

This request is used to update the data of an existing Contact.

Requirements

This request can only be executed if the Contact already exists.

Features

- Mandatory fields must always be filled in, even if they remain unchanged.
- Omitting any keywords when applying the key/value format has the effect of deleting the values previously set for them.
- If a Contact is used as Holder in a domain which has a DISPUTE entry placed on it, you cannot use the UPDATE request to change all of its data. Name, organization, address, postal code, city and country cannot be updated. The RegAcc must contact Business Services to have the changes implemented.

Verification

- Verification information is added, changed or removed from an existing contact of the type PERSON or ORG.
- Verification information can be deleted by not providing any information during an update.

- A Contact UPDATE request triggers a new verification process for a domain, and the syntax and completeness check and risk assessment are carried out again.
- The contact handles of the domain portfolio are not systematically scanned to determine a risk assessment for all domains. This does not apply to event-driven reviews of individual portfolio domains.
- Should an event-driven review reveal a suspicious or very high risk, a verification must be carried out as part of a domain request, which means an update of the contact.
- Verification information can also be created or modified for a contact for a domain that is subject to a DISPUTE procedure.

Contact Type

DThe contact type can be changed with an UPDATE:

- from PERSON to ORG or
- from ORG to PERSON.

> **Notice**
>
> The contact type cannot be changed for contacts that are used for a domain on which a DISPUTE is set.

Request Parameters

A Contact UPDATE request is composed of the "Contact" data object fields and some other parameters, which are described below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	contact:update	1	enumeration	update-erule	request type
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/value format
CtId	ctid	0 -1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

The Contact handle stated in the request does not exist.

Data object PERSON and ORG

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<MemberID >	• A Contact handle must always start with the RegAccId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	These Contact types are available: • PERSON = natural person, • ORG = a legal person (company, association, grouping of holders, organization etc., such as "DENIC eG")	• The type of contact can be changed later. • For Holder you may only use the types PERSON or ORG.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Name	contact:name	1	nor-malizedString 1 - 255	name-rule	This is the Contact's name.	• Once set, you cannot change the keyword "name" any more.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Organisation	con-tact:organisation	0 - *	nor-malizedString 1 - 255	organisation-rule	This is the organ-isation which is represented by the Contact.	• You may only use the types PERSON. • Cannot be changed, if the domain bears a DISPUTE entry. • You must always state the domain holder in the "Name:" field. The "Organ-ization" value

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						has no right to the domain and is not entitled to apply for an AuthInfo or a provider change or similar.
Address	contact:address	1 - *	nor-malizedString 1 - 255	address-rule	Street and house number of Contact	The keyword cannot be changed if the Contact is ref-erenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
PostalCode	contact:postalCode	1	token 1 - 20	postalcode-rule	Postal code and address of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.
City	contact:city	1	normalizedString 1 - 80	city-rule	Place of residence of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
CountryCode	con-tact:countryCode	1	Enumeration 2	country-erule	Country code of the country in which the Contact's place of residence is loc-ated	• You must use the country code of the ISO-3166-1 alpha-2 list ISO-3166-1 alpha-2 Liste. • The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						entry.
Email	contact:email	1 - *	nor-malizedString 3 - 255	email-rule (see RFC5322 - Internet Message Format)	E-mail address of Contact	„email" always is a mandatory field and must be specified at least once for the types PERSON and ORG in Contact CREATE and Contact UPDATE requests

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
[VerificationInformation]	<verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:-	-	-	Once per verification record	-	-	For K/V: header line that precedes the verific-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	type- ="ver- ific- ation:veri- ficationInformationType"> <verification:verifiedClaims> <veri- fication:verificationResult> <veri-						ation record

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fication:verificationReference> <veri- fic- ation:verificationTimestamp> <veri- fication:verificationEvidence> <veri- fication:verificationMethod> </veri-						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fic-ation:verificationInformation>						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
-	-	<veri-fic-ation:verifiedClaims> <verification:claim> </veri-fic-ation:veri-fiedClaims>	-	-	-	-	-

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerifiedClaim	-	-	<verification:claim> ... <verification:claim>	1 - 3	normalizedString / fixed length due to predefined valued	claim-rule	"claims" are data that has been verified **Special features for e-mail** The

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
							complete verification information block with all keywords or XML elements and val-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							must be spe-cified for an email, if the res-ult of the email check was neg-ative, in other words,

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							A sub-sequent update with the positive value ‘suc-cess’ must also be com-munic-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							ated, after a fail. If the value 'suc-cess' was present at the start of the check, the veri-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							fication inform-ation block for email **can** be repor-ted.

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationResult	-	<verification:verificationResult> ... </verification:verificationResult>	-	1	normalizedString / fixed length due to predefined valued	result-rule	Notification of the verification result

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationReference	-	<verification:verificationReference> ... /verification:verificationReference>	-	1	normalizedString / length is defined by member	reference-rule	The content is free text that makes reference to a request number or order number, etc.

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationTimestamp	-	<verification:verificationTimestamp> ... </verification:verification	-	1	dateTime	timestamp-rule	Point in time, when the verification was performed

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
		fic- ationTimestamp>					

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationEvid-ence	-	<veri-fic-ation:veri-ficationEvidence> ... </veri-fic-ation:veri-ficationEvidence>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	evid-ence-rule	Evid-ence that was checked within the scope of the veri-fication (e.g. for the value "idcar-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							d", which cor-respon-ds to identity card)

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationMethod	-	<veri-fic-ation:veri-ficationMethod> ... </veri-fic-ation:veri-ficationMethod>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	meth-od-rule	Method used to carry out the veri-fication (e.g. for the value "pvr", which stands for evid-encing via "video

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							iden-tific-ation")

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
TrustFramework	-	<verification:trustFramework> ... </verification:trustFramework>	-	1	normalizedString / fixed length due to predefined valued	framework-rule	Framework used for the verification At present, only the value "de_denic" exists, others may follow

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							later.

Data object REQUEST

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<Mem-berID >	A Contact handle must always start with the RegAc-cId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	REQUEST = an e-mail address or a URL in the URI-Template format.	For General Request and Abuse Contact you may only use the type REQUEST.
URI-Tem-plate	contact:uri-tem-plate	1	normalizedString 8 - 1024	Syntax as defined in RFC 6570 - URI Tem-plate	The variables Ula-bel and Alabel in the URI-Template are replaced with the domain at a domain query (web-whois).	In case of a CREATE request, the content of the URI-Template is converted into a URL or e-mail as a test (For Ala-bel and Ulabel, a sample domain is used.) to check the syntax on cor-rectness.

Examples

- xml
- k/v

format: xml

request: contactUPDATE

type: PERSON

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:verification="http://registry.denic.de/verification/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <contact:update>
    <contact:handle>DENIC-1000022-EXAMPLE-XML-PERSON</contact:handle>
    <contact:type>PERSON</contact:type>
    <contact:name>John Doe</contact:name>
    <contact:organisation>DENIC eG</contact:organisation>
    <contact:postal>
      <contact:address>Theodor-Stern-Kai 1</contact:address>
      <contact:postalCode>60596</contact:postalCode>
      <contact:city>Frankfurt am Main</contact:city>
      <contact:countryCode>DE</contact:countryCode>
    </contact:postal>
    <contact:email>john.doe@denic.de</contact:email>
```

format: xml**request:** contactUPDATE**type:** PERSON

```
    <contact:phone>+49.69272350</contact:phone>
    <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0"
xsi:type="verification:verificationInformationType">
      <verification:verifiedClaims>
       <verification:claim>name</verification:claim>
       <verification:claim>address</verification:claim>
      </verification:verifiedClaims>
      <verification:verificationResult>success</verification:verificationResult>
      <verification:verificationReference>ABC123/45GHT</verification:verificationReference>
      <verification:verificationTimestamp>2023-11-11T15:36:21+02:00</verification:verificationTimestamp>
      <verification:verificationEvidence>idcard</verification:verificationEvidence>
      <verification:verificationMethod>auth</verification:verificationMethod>
      <verification:trustFramework>de_denic</verification:trustFramework>
    </verification:verificationInformation>
  </contact:update>
  <ctid>xml-74ba5118</ctid>
</registry-request>
```

format: xml

response: contactUPDATE

type: PERSON

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="h-
ttp://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>25cf1349-dfb4-48e9-951c-af60b068ba24</tr:stid>
    <tr:ctid>xml-74ba5118</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: xml

request: contactUPDATE

type: REQUEST

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```
<registry-request xmlns="https://registry.denic.de/global/5.0" xmlns:contact="https://registry.denic.de/contact/5.0">
  <contact:update>
    <contact:handle>DENIC-99995-GENREQ</contact:handle>
    <contact:type>REQUEST</contact:type>
    <contact:uri-template>https://denic.de/contact/form{?Alabel,Ulabel}</contact:uritemplate>
  </contact:update>
  <ctid>cba-987654321</ctid>
</registry-request>
```

format: xml

response: contactUPDATE

type: REQUEST

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>bb135322-958c-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
```

```
</registry-response>
```

format: k/v

request: contactUPDATE

type: PERSON

```
Version: 5.0
Action: CREATE
Handle: DENIC-1000022-EXAMPLE-PERSON
Type: PERSON
Organisation: DENIC eG
Name: Max Mustermann
Address: Business Services
Address: Theodor-Stern-Kai 1
Address: in Frankfurt
Address: south side
Address: riverside
PostalCode: 60596
City: Frankfurt am Main
CountryCode: DE
eMail: email-1@denic.de
eMail: email-2@denic.de
eMail: email-3@denic.de
eMail: email-4@denic.de
eMail: email-5@denic.de
```

```
eMail: email-6@denic.de
Phone: +49.6927235x290
CTID: kv-7bf04fa8

[VerificationInformation]
VerifiedClaim: name
VerifiedClaim: address
VerificationResult: success
VerificationReference: ABC123/45GHT
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: idcard
VerificationMethod: auth
TrustFramework: de_denic

[VerificationInformation]
VerifiedClaim: email
VerificationResult: failed
VerificationReference: 354546TZQ
VerificationTimestamp: 2023-10-04T12:22:19+02:00
VerificationEvidence: email_ver_transaction_log
VerificationMethod: reachability
TrustFramework: de_denic
```

format: k/v

response: contactUPDATE

type: PERSON

```
RESULT: success
STID: c5f61739-600d-4c3c-b0a7-058bf8254aa7
CTID: kv-7bf04fa8
```

format: k/v

request: contactUPDATE

type: REQUEST

```
Version: 5.0
Action: update
Type: request
Handle: DENIC-99995-GENREQ
Uri-template: https://denic.de/contact/form{?Alabel,Ulabel}
```

format: k/v

response: contactUPDATE

type: REQUEST

```
Result: success
STID: 4cfcf3ff-958c-11df-a9e2-9519b5688882
```

Replies to contactCREATE and contactUPDATE Requests

Description of Data Fields

The replies of the .de registration system to Contact CREATE or Contact UPDATE requests comprise the data fields listed below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
StId	tr:stid	1	token 30 - 255	stid-rule	This is a Server Transaction ID (StId) which is assigned by the server. The StId ensures that the request from the provider and DENIC's reply are referenced with a globally unique ID.
CtId	tr:ctid	0 - 1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	tr:result	1	enumeration	result-erule	This is the processing result: • success = request was executed. • failed = request was not executed. • exception = system failure. Request was not executed.
Info	tr:message level-l="info"	0 - *	normalizedString 1 - 255	message-rule	"info" provides information related to the request. It includes the message code and text. You will find more detailed information in the Message Code section.
Warning	tr:message level-l="warning"	0 - *	normalizedString 1 - 255	message-rule	These are warnings related to the request. They include the message code and text.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Error	tr:message level-l="error"	0 - *	normalizedString 1 - 255	message-rule	These are error messages related to the request. They include the mes-sage code and text.

contactCHECK Request

Function

This request is used to query the status of a Contact.

Requirements

None

Features

You can only query the status of Contacts administered by your own.

Request Parameters

A Contact CHECK request comprises the parameters described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	contact:check	1	enumeration	check-erule	request type
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/-value format.
Handle	contact:handle	1	enumeration	contact-rule	This is the unique ID of the Contact. Apply the following syntax: <RegAccId>-<ID of DENIC member>

Examples

- xml
- k/v

format: xml

request: contactCHECK

```
<registry-request xmlns="http://registry.denic.de/global/5.0"
xmlns:contact="http://registry.denic.de/contact/5.0">
  <contact:check>
    <contact:handle>DENIC-1000002-MAX</contact:handle>
  </contact:check>
</registry-request>
```

format: xml

response: contactCHECK

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>ba1b8093-9a2b-11df-a9e2-9519b5688882</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
      <contact:checkData xmlns:contact="h-
ttp://registry.denic.de/contact/5.0">
        <contact:handle>DENIC-1000002-MAX</contact:handle>
        <contact:status>exist</contact:status>
      </contact:checkData>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: k/v

request: contactCHECK

```
Version: 5.0
Action: check
Handle: DENIC-1000002-MAX
```

format: k/v

response: contactCHECK

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

Handle: DENIC-1000002-MAX
Status: exist
```

Replies to contactCHECK Requests

Description of Data Fields

The replies of the .de registration system to Contact STATUS requests comprise the data fields listed below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
StId	tr:stid	1	token 30 - 255	stid-rule	This is a Server Transaction ID (StId) which is assigned by the server. The StId ensures that the request from the provider and DENIC's reply are referenced with a globally unique ID.
CtId	tr:ctid	0 - 1	token 3 - 64	Any visible Unicode char-acter (according to Unicode ver-sion 3.1)	This is the unique transaction ID which is issued by the client.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	tr:result	1	enumeration	result-erule	This is the processing result: • success = request was executed. • failed = request was not executed. • exception = system failure. Request was not executed.
Info	tr:message level-l="info"	0 - *	normalizedString 1 - 255	message-rule	"info" provides information related to the request. It includes the message code and text.
Warning	tr:message level-l="warning"	0 - *	normalizedString 1 - 255	message-rule	These are warnings related to the request. They include the mes-sage code and text.
Error	tr:message level-l="error"	0 - *	normalizedString 1 - 255	message-rule	These are error messages related to the request. They include the message code and text.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact. Apply the following syntax: <RegAccId>-<ID of DENIC member>
Status	contact:status	1	enumeration	contactstatus-erule	"status" provides information about the status of the Contact. • free = Contact does not exist • exists = Contact exists • invalid = Contact is invalid

contactINFO Request

Function

This request is used to query the data of a Contact.

Requirements

The Contact must exist.

Features

You can only query data of Contacts administered by your own.

Request Parameters

A request comprises the parameters described below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	contact info	1	enumeration	Info-erule	request type
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/value format.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact. Apply the following syntax: <RegAccId>-<ID of DENIC member>

Common Mistakes

The Contact handle stated in the request does not exist.

Data object for PERSON and ORG

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<MemberID >	• A Contact handle must always start with the RegAccId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	These Contact types are available: • PERSON = natural person, • ORG = a legal person (company, association, grouping of holders, organization etc., such as "DENIC eG")	• The type of contact can be changed later. • For Holder you may only use the types PERSON or ORG.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Name	contact:name	1	nor-malizedString 1 - 255	name-rule	This is the Contact's name.	• Once set, you cannot change the keyword "name" any more.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Organisation	con-tact:organisation	0 - *	nor-malizedString 1 - 255	organisation-rule	This is the organ-isation which is represented by the Contact.	• You may only use the types PERSON. • Cannot be changed, if the domain bears a DISPUTE entry. • You must always state the domain holder in the "Name:" field. The "Organ-ization" value

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						has no right to the domain and is not entitled to apply for an AuthInfo or a provider change or similar.
Address	contact:address	1 - *	nor-malizedString 1 - 255	address-rule	Street and house number of Contact	The keyword cannot be changed if the Contact is ref-erenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
PostalCode	contact:postalCode	1	token 1 - 20	postalcode-rule	Postal code and address of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.
City	contact:city	1	normalizedString 1 - 80	city-rule	Place of residence of Contact	The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE entry.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
CountryCode	con-tact:countryCode	1	Enumeration 2	country-erule	Country code of the country in which the Contact's place of residence is loc-ated	• You must use the country code of the ISO-3166-1 alpha-2 list ISO-3166-1 alpha-2 Liste. • The keyword cannot be changed if the Contact is referenced as the holder of a domain which bears a DISPUTE

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						entry.
Email	contact:email	1 - *	nor-malizedString 3 - 255	email-rule (see RFC5322 - Internet Message Format)	E-mail address of Contact	„email" always is a mandatory field and must be specified at least once for the types PERSON and ORG in Contact CREATE and Contact UPDATE requests

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
[VerificationInformation]	<verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:-	-	-	Once per verification record	-	-	For K/V: header line that precedes the verific-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	type- ="ver- ific- ation:veri- ficationInformationType"> <verification:verifiedClaims> <veri- fication:verificationResult> <veri-						ation record

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fication:verificationReference> <veri- fic- ation:verificationTimestamp> <veri- fication:verificationEvidence> <veri- fication:verificationMethod> </veri-						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
	fic-ation:verificationInformation>						

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
-	-	<veri-fic-ation:verifiedClaims> <verification:claim> </veri-fic-ation:veri-fiedClaims>	-	-	-	-	-

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerifiedClaim	-	-	<verification:claim> ... <verification:claim>	1 - 3	normalizedString / fixed length due to predefined valued	claim-rule	"claims" are data that has been verified **Special features for e-mail** The

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
							complete verification information block with all keywords or XML elements and val-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							must be spe-cified for an email, if the res-ult of the email check was neg-ative, in other words,

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							A sub-sequent update with the positive value 'suc-cess' must also be com-munic-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							ated, after a fail. If the value 'suc-cess' was present at the start of the check, the veri-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							fication inform-ation block for email **can** be repor-ted.

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationResult	-	<verification:verificationResult> ... </verification:verificationResult>	-	1	normalizedString / fixed length due to predefined valued	result-rule	Notification of the verification result

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationReference	-	<verification:verificationReference> ... /verification:verificationReference>	-	1	normalizedString / length is defined by member	reference-rule	The content is free text that makes reference to a request number or order number, etc.

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationTimestamp	-	<verification:verificationTimestamp> ... </verification:verification	-	1	date-time	timestamp-rule	Point in time, when the verification was performed

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
		fic- ationTimestamp>					

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
Veri-fic-ationEvid-ence	-	<veri-fic-ation:veri-ficationEvidence> ... </veri-fic-ation:veri-ficationEvidence>	-	1	nor-mal-izedStrin-g / fixed length due to pre-defined valued	evid-ence-rule	Evid-ence that was checked within the scope of the veri-fication (e.g. for the value "idcar-

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							d", which cor-respon-ds to identity card)

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
VerificationMethod	-	<verification:verificationMethod> ... </verification:verificationMethod>	-	1	normalizedString / fixed length due to predefined valued	method-rule	Method used to carry out the verification (e.g. for the value "pvr", which stands for evidencing via "video

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							iden-tific-ation")

K/V Keyword	XML Namespace and Elements	1st Nesting	2nd Nesting	Occurrence, min – max per verification information block	Type / Length	Value Range	Description
TrustFramework	-	<verification:trustFramework> ... </verification:trustFramework>	-	1	normalizedString / fixed length due to predefined valued	framework-rule	Framework used for the verification At present, only the value "de_denic" exists, others may follow

K/V Key-word	XML Namespace and Elements	1st Nesting	2nd Nest-ing	Occur-rence, min – max per veri-fic-ation infor-matio-n block	Type / Length	Value Rang-e	Descrip-tion
							later.

Data object REQUEST

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact.Syntax: <RegAccId>-<Mem-berID >	A Contact handle must always start with the RegAc-cId of the admin-istering RegAcc.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Type	contact:type	1	enumeration	role-erule	REQUEST = an e-mail address or a URL in the URI-Template format.	For General Request and Abuse Contact you may only use the type REQUEST.
URI-Tem-plate	contact:uri-tem-plate	1	normalizedString 8 - 1024	Syntax as defined in RFC 6570 - URI Tem-plate	The variables Ula-bel and Alabel in the URI-Template are replaced with the domain at a domain query (web-whois).	In case of a CREATE request, the content of the URI-Template is converted into a URL or e-mail as a test (For Ala-bel and Ulabel, a sample domain is used.) to check the syntax on cor-rectness.

Examples

- xml
- k/v

format: xml

request: contactINFO

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:contact="h-
ttp://registry.denic.de/contact/5.0" xmlns:verification="http://registry.denic.de/verification/5.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <contact:info>
    <contact:handle>DENIC-1000022-EXAMPLE-XML-PERSON</contact:handle>
  </contact:info>
</registry-request>
```

format: xml

response: contactINFO

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

format: xml**response:** contactINFO

```
2
3  <registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="h-
   ttp://registry.denic.de/transaction/5.0">
4    <tr:transaction>
5      <tr:stid>b61d054e-d751-4c96-a180-c2eb4230a374</tr:stid>
6      <tr:result>success</tr:result>
7      <tr:data>
8        <contact:infoData xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:xsi="h-
   ttp://www.w3.org/2001/XMLSchema-instance">
9          <contact:handle>DENIC-1000022-EXAMPLE-XML-PERSON</contact:handle>
10         <contact:type>PERSON</contact:type>
11         <contact:name>John Doe</contact:name>
12         <contact:organisation>DENIC eG</contact:organisation>
13         <contact:postal>
14           <contact:address>Theodor-Stern-Kai 1</contact:address>
15           <contact:postalCode>60596</contact:postalCode>
16           <contact:city>Frankfurt am Main</contact:city>
17           <contact:countryCode>DE</contact:countryCode>
18         </contact:postal>
19         <contact:email>john.doe@denic.de</contact:email>
20         <contact:phone>+49.69272350</contact:phone>
21         <contact:changed>2024-09-27T16:40:53+02:00</contact:changed>
22         <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0"
   xsi:type="verification:verificationInformationType">
23           <verification:verifiedClaims>
24             <verification:claim>address</verification:claim>
25             <verification:claim>name</verification:claim>
```

format: xml**response:** contactINFO

```
            </verification:verifiedClaims>
            <verification:verificationResult>success</verification:verificationResult>
            <verification:verificationReference>ABC123/45GHT</verification:verificationReference>
            <verification:verificationTimestamp>2023-11-11T14:36:21+01:00</verification:verificationTimestamp>
            <verification:verificationEvidence>idcard</verification:verificationEvidence>
            <verification:verificationMethod>auth</verification:verificationMethod>
            <verification:trustFramework>de_denic</verification:trustFramework>
          </verification:verificationInformation>
        </contact:infoData>
      </tr:data>
    </tr:transaction>
</registry-response>
```

format: k/v

request: contactINFO

```
Version: 5.0
Action: INFO
Handle: DENIC-1000022-EXAMPLE-PERSON
```

format: k/v

response: contactINFO

type: PERSON

```
RESULT: success
STID: 8e9ccb16-f362-4eb3-9428-763bae69705b

Handle: DENIC-1000022-EXAMPLE-PERSON
Type: PERSON
Organisation: DENIC eG
Name: Max Mustermann
Address: Business Services
Address: Theodor-Stern-Kai 1
Address: in Frankfurt
Address: south side
Address: riverside
PostalCode: 60596
City: Frankfurt am Main
CountryCode: DE
eMail: email-1@denic.de
eMail: email-2@denic.de
eMail: email-3@denic.de
eMail: email-4@denic.de
eMail: email-5@denic.de
eMail: email-6@denic.de
Phone: +49.6927235x290
Changed: 2024-09-27T16:38:17+02:00
```

format: k/v**response:** contactINFO**type:** PERSON

```
24
25 [VerificationInformation]
26 VerifiedClaim: name
27 VerifiedClaim: address
28 VerificationResult: success
29 VerificationReference: ABC123/45GHT
30 VerificationTimestamp: 2023-11-11T15:36:21+02:00
31 VerificationEvidence: idcard
32 VerificationMethod: auth
33 TrustFramework: de_denic
34
35 [VerificationInformation]
36 VerifiedClaim: email
37 VerificationResult: failed
38 VerificationReference: 354546TZQ
39 VerificationTimestamp: 2023-10-04T12:22:19+02:00
40 VerificationEvidence: email_ver_transaction_log
41 VerificationMethod: reachability
42 TrustFramework: de_denic
```

Replies to contactINFO Requests

Important Information

When querying a handle of type REQUEST, the variables in the URI-Template are not replaced with domains.

Description of Data Fields

The replies of the .de registration system to Contact INFO requests comprise the data fields described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
StId	tr:stid	1	token 3 - 64	stid-rule	This is a Server Transaction ID (StId) which is assigned by the server. The StId ensures that the request from the provider and DENIC's reply are referenced with a globally unique ID.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	tr:result	1	enumeration	result-erule	This is the processing result: • success = request was executed. • failed = request was not executed. • exception = system error. Request was not executed.
Info	tr:message level="info"	0 - *	normalizedString 1 – 255	message-rule	"info" provides information related to the request. It includes the message code and text.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Warning	tr:message level-l="warning"	0 - *	normalizedString 1 – 255	message-rule	These are warnings related to the request. They include the message code and text.
Error	tr:message level-l="error"	0 - *	normalizedString 1 – 255	message-rule	This is an error message related to the request. It includes the message code and text.
Handle	contact:handle	1	token 9 - 32	contact-rule	This is the unique ID of the Contact. Apply the following syntax: <RegAccId>-<ID of DENIC member>

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Type	contact:type	1	Enumeration	role-erule	"type" defines the type of the Contact: • PERSON = natural person • ORG = a legal person (company, association, grouping of holders, organization etc., e.g. "DENIC eG".) • REQUEST = URL or Mailto link (in form of a URI-Template)

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
					for General Request or Abuse Contact
Name	contact:name	1	normalizedString 1 – 255	name-rule	This is the Contact's name.
Organisation	contact:organisation	0 - *	normalizedString 1 – 255	organisation-rule	This is the organization which is represented by the Person.
Address	contact:address	1 - *	normalizedString 1 – 255	address-rule	Street and house number
PostalCode	contact:postalCode	1	token 1 – 20	postalcode-rule	This is the postal code of the address. NO country code is given.
City	contact:city	1	normalizedString 1 – 80	city-rule	This is the place of residence.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
CountryCode	contact:countryCode	1	enumeration	country-erule	This is the country code of the country in which the place of residence is situated.
Email	contact:email	1 - *	token 1 – 255	email-rule	This is the e-mail address.
DisputeReference	disputeReference	0 - 1	Boolean	boolean-erule	The function of the keyword "disputereference" is to inform whether a Contact is used as domain holder refers to a domain that bears a DISPUTE entry. If such reference is made, the keyword has the value "true".
Changed	changed	1	dateTime 25	date-rule	This is the time stamp of the most recent change in the record concerned.

Error Messages Related to Contact Requests

Potential Error Messages

Below you will find a list of the error messages that will occur exclusively in relation with Contact requests.

For general error messages which may occur in relation with Contact and also with other requests, refer to the Annex.

Comments on the Table

- The column "Format of incoming message" informs whether the error message may occur with the key/value (= K/V) and / or with the XML format (= XML).
- The column "Create / Update" informs whether the error message may occur when creating (= C) and / or when changing (= U) a Contact.

Error Code	Error Message	Output Format	Create/ Update	Brief Description
53000080002	This value cannot be deleted	K/V, XML	U	For certain functions, you must state specific obligatory data. Theses values must not be deleted.
63200020603	This Contact is administered by another provider [dynamic error text]	K/V, XML	U	The Contact is administered by another provider.
63200040610	Keyword "Handle" cannot be recognized	K/V	C, U	The keyword "Handle" either was omitted, not placed at the start of its line or did not have a value assigned to it.

Error Code	Error Message	Output Format	Create/ Update	Brief Description
63200040702	Keyword "Type" must appear once and just once	K/V	C, U	The keyword "Type" occurs more than once or the value assigned to it includes a line feed.
63200040710	Keyword "Type" cannot be recognized	K/V	C, U	The keyword "Type" either was omitted, not placed at the start of its line or did not have a value assigned to it.
63200040802	Keyword "Name" must appear once and just once	K/V	C, U	The keyword "Name" occurs more than once or the value assigned to it includes a line feed.
63200040810	Keyword "Name" cannot be recognized	K/V	C, U	The keyword "Name" either was omitted, not placed at the start of its line or did not have a value assigned to it.
63200041010	Keyword "Address" cannot be recognized	K/V	C, U	The keyword "Address" either was omitted, not placed at the start of its line or did not have a value assigned to it.

Error Code	Error Message	Output Format	Create/ Update	Brief Description
63200041102	Keyword "PostalCode" must appear once and just once	K/V	C, U	The keyword "PostalCode" occurs more than once or the value assigned to it includes a line feed.
63200041110	Keyword "PostalCode" cannot be recognized	K/V	C, U	The keyword "PostalCode" either was omitted, not placed at the start of its line or did not have a value assigned to it.
63200041202	Keyword "City" must appear once and just once	K/V	C, U	The keyword "City" occurs more than once or the value assigned to it includes a line feed.
63200041210	Keyword "City" cannot be recognized	K/V	C, U	The keyword "City" either was omitted, not placed at the start of its line or did not have a value assigned to it.
63200041302	Keyword "CountryCode" must appear once and just once	K/V	C, U	The keyword "CountryCode" occurs more than once or the value assigned to it includes a line feed.

Error Code	Error Message	Output Format	Create/ Update	Brief Description
63200041310	Keyword "CountryCode" cannot be recognized	K/V	C, U	The keyword "CountryCode" either was omitted, not placed at the start of its line or did not have a value assigned to it.
63200041610	Keyword "Email" cannot be recognized	K/V, XML	C, U	Keyword "Email" does not exist, does not appear at the beginning of the line, or the value is empty.
63200060702	Inadmissible value for keyword "Type"	K/V	C, U	The value entered for the keyword "Type" is invalid. You must use "PERSON", "ORG" or “REQUEST”.
63200061302	The value for "CountryCode" is not a valid ISO code	K/V, XML	C, U	The value assigned to the keyword "CountryCode" must correspond to an ISO country code (ISO-3166-1 alpha-2).
63200061603	The value for keyword "Email" has an invalid format	K/V, XML	C, U	The value for "Email" has an invalid format.

Error Code	Error Message	Output Format	Create/ Update	Brief Description
63200080900	Keyword "Organisation" invalid with "Type" "ORG"	K/V, XML	C, U	The keyword "Organisation" may only be included if the value assigned to "Type" is "PERSON".
63220080701	The value of "Type" cannot be changed with an UPDATE	K/V, XML	U	Once set, the value of the keyword "Type" is immutable and cannot be changed later.
63220080801	The value of "Name" cannot be changed with an UPDATE	K/V, XML	U	Once set, the value of the keyword "Name" is immutable and cannot be changed later.
83000000005	Validation of URI-Template failed [„empty URI-Template“]	K/V, XML	C	The URI-Template string must not be empty.
83000000005	Validation of URI-Template failed [URI-Template too long:max length allowed is 1024]	K/V, XML	C, U	The URI-Template string must not exceed 1024 characters in length.

Error Code	Error Message	Output Format	Create/ Update	Brief Description
83000000005	Validation of URI-Template failed [invalid URI scheme (1)]	K/V, XML	C, U	**(1)** The incorrect URI scheme is listed in the error message. Only the URI schemes http, https and mailto are supported.
83000000005	Validation of URI-Template failed [invalid URI-Template variable(s) [(2)]]	K/V, XML	C, U	**(2)** The invalid variable name is repeated in the error message. Only variables of the **Alabel** and **Ulabel** notation are supported. The variables are case-sensitive. URI-Template strings without variables can also be used. The URI-Template string may contain both types of variables. Unknow variables will always trigger an error. The variable value representing a domain is shown in an ASCII code compatible format for **Alabel** and in Unicode format for **Ulabel**.

Error Code	Error Message	Output Format	Create/ Update	Brief Description
83000000005	Validation of URI-Template failed [The variable name (3) contains invalid characters]	K/V, XML	C, U	**(3)** Display of variable name including the invalid characters.

Requests Related to the Domain Inventory

Overview

Introduction

Requests related to domains are used to link contacts and contents to the domain concerned. Before you can use a Contact handle in domain requests, the handle must be created by the administering RegAcc.

Contents

This section covers the following topics:

- Domain Data Object
- Domain CREATE Request
- Domain UPDATE Request
- Domain CHHOLDER Request
- Domain DELETE Request
- Domain RESTORE Request
- Domain TRANSIT Request
- Reply to Domain CREATE, UPDATE, CHHOLDER, RESTORE, DELETE, TRANSIT Requests
- Domain CHECK Request
- Domain INFO Request

- Reply to Domain CHECK / INFO Requests
- Error Messages Related to Domain Requests

Important Information

All the Contact handles used in domain requests must be created by the administering RegAcc and must still exist.

Domain Statuses pendingCreate and serverHold

pendingCreate

The status indicates that a Domain CREATE request has been executed for the domain, but that no risk assessment has yet been carried out.

Properties:

- The domain is registered.
- The domain is not delegated.
- The status is set after a domain CREATE request when the risk assessment is still pending.
- The status after that is either “connect” or “serverHold”; the status is communicated by message or email.
- “pendingCreate“ cannot be set in combination with any other status ("free", "connect", "failed", "redemptionPeriod", "serverHold").

serverHold

The status is set by the registry. The domain is registered but is not published in the DNS (de-delegated).

Properties:

- The domain is registered.
- The domain is de-delegated.
- The status is set when the domain is in quarantine.

- “serverHold“ cannot be set in combination with another status ("free", "connect", "failed", "redemptionPeriod", "serverHold").

Domain Data Object

Contents of the Data Object

The "Domain" data object comprises all the relevant administrative and technical data of a domain.

Processing Options

A domain can be registered and deleted or restored. Domain data can be updated.

With a TRANSIT request, you can transfer domain administration to DENIC.

Domain Data Objects: Syntax and Processing Rules

The table below informs about the syntax of the "Domain" data object.

Additionally, you will find rules you must observe when processing the "Domain" data object.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Domain	domain:handle	1	token 4 - 66	domain-rule	This is the domain name.	-

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Domain-Ace	domain:ace	0 - 1	token 4 - 66	domainace-rule	This is the ACE form (ASCII Compatible Encoding) of the domain name.	-
Holder	domain:contact role-e="holder"	1 - 5	token 9 - 32	contact-rule	This is the handle ID of the domain holder.	Only Contacts of the PERSON or ORG type are permitted. If several domain holders exist, all the handle IDs must be different.
GeneralRequest	domain:contact role-e="generalrequest"	0-1	token 9 - 32	contact-rule	Handle-Id of General Request	Only contacts of the REQUEST types are permitted
AbuseContact	domain:contact role-e="abusecontact"	0-1	token 9 - 32	contact-rule	Handle-Id of Abuse Contact	Only contacts of the REQUEST types are permitted.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Name server records						
Nserver	dnsentry:owner	2 - 13	token 1 - 254	nserver-rule	If you use the k/v format: host name of name server If you use the XML format: ACE form of domain	The name servers must meet the requirements of the Predelegation Checks (see documentation "Nameserver Predelegation Check").
-	dnsentry:-nameserver	0-13	token 1 - 254	nserver-rule	This is the host name of the name server.	For name servers whose host name is located within the domain for which they make available the DNS, you must state an IPv4 address. Indication of an IPv6 address is optional.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
-	dnsentry:address	0-13	token 7-39	ip-rule IPv4 addresses according to RFC791 - INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION in „dotted-quad“ format	This is the IPv4 address of the name server	You must state the IPv4 address.
-	dnsentry:addressV6	0-8	token 7-39	ip-rule IPv6 addresses according to RFC4291 IP Version 6 Addressing Architectureas „preferred form“.	This is the IPv6 address of the name server	You must state the IPv6 address.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy

Dnskey	dnsentry:flags
	dnsentry:protocol
	dnsentry:algorithm
	dnsentry:publicKey
		0-20	token
			30 - 999
				dnskey-rule
				
				Syntax according to RFC4034 - Resource Records for the DNS Security Extensions, DNSKEY as „presentation format“
					Dnskey	The following four values must be stated:
						• Flags: 256 or 257
						• Protocol: at present always 3
						• Numeric encoding of algorithm that is used:
						  • 3 (DSA/SHA-1)
						  • 5 (RSA/SHA-1)

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						• 6 (DSA-NSEC3-SHA1) • 7 (RSASHA-1-NSEC3-SHA1) • 8 (RSA/SHA-256) • 10 (RSA/SHA-512) • 12 (GOST) • 13

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						(ECDSA-256) • 14 (ECDSA-512) • Public key: base64-encoded • You will find descriptions of the DNSSEC-specific Predelegation Checks in the documentation "Nameserver

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
						Predelegation Check"
Alternative: Nsentries						
IN A record:						

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Nsentry	dnsentry:owner	1 - 5	token 4 - 252	nsentry-rule	This is a domain which shall be registered or a sub-domain of the domain to be registered, including the ".de"	State in ACE form

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
	dnsentry:address		token 3 - 39	ip-rule	This is the IPv4/IPv6 address of the host Consecutive colons (::) as unspecified address are supported and resolved before starting the length check.	IPv4 addresses according to RFC791 - INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION in „dotted-quad“ format IPv6 addresses according to RFC4291 - IP Version 6 Addressing Architecture in „preferred form“
IN MX-record:						

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Nsentry	dnsentry:owner	1 - 5	token 4 - 252	nsentry-rule	This is the domain or the sub-domain: You may enter a wildcard (*.domain.de) when MX-records are used. Length includ-ing the ".de"	State in ACE form
	dnsentry:preference		unsignedShort	mxpref-rule	This is the MX preference value.	0 = highest preference to 999 = lowest pref-erence
	dnsentry:exchange		token 4 - 252	mx-rule	This is the name of the relevant mail server. Length includ-ing the ".de"	State in ACE form . for a "Null MX" No Service Resource Record for Domains That Accept No Mail

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description	Policy
Status	domain:status	1	enumeration	status-erule	This is the status of the domain.	Will be automatically assigned by the system.
Expire	domain:expire	0 – 1	dateTime 25	date-rule	This is the date, when a domain with the "failed" status is deleted.	Will be automatically assigned by the system.

domainCREATE Request

Function

This request is used to register a domain.

Requirements

- The request can only be processed if the domain has not already been registered.
- Before registering the domain you must create the Contacts that are referenced in the request.

Verification

- Domains in the “failed” state are initially bound to the 30-day connection period. After the connection, the risk assessment is carried out.
- If a contact handle (type “PERSON” or “ORG”) with verified contact data is used, every domain registered with a domain CREATE request is automatically verified, provided that the verification result of the contact remains unchanged as “success”.
- A domain CREATE request is also executed if contacts without a phone number or email address are used.
- When the domain is passed to the risk assessment without one of the final statuses “connect” or “serverHold”, the domain status “pendingCreate” is set. An information message is output in the response.

Request Parameters

A Domain CREATE request is composed of the fields of the "Domain" data object and some other parameters, which are described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	domain:create	1	enumeration	createerule	This is the request type.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

- The requested domain already exists.
- The status of the requested domain is „redemptionPeriod“.
- One of the Contacts stated in the request does not meet the requirements of the role to be assigned to it.
- The Nameserver Predelegation Check failed. You will find descriptions of the checks that are executed in the document "Nameserver Predelegation Check".

Examples

- xml
- k/v

format: xml

format: xml**request:** domainCREATE

request: domainCREATE

format: xml

response: domainCREATE

status: pendingCreate

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>a503e644-635c-4253-b598-84d0c2296a97</tr:stid>
    <tr:ctid>xml-74ba5155</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080014">
      <tr:text>Domain "Status" is "pendingCreate"</tr:text>
    </tr:message>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainCREATE

```
Version: 5.0
Action: CREATE
Domain: domain-example-nserver.de
Holder: DENIC-1000022-EXAMPLE-PERSON
Generalrequest: DENIC-1000022-EXAMPLE-GR
Abusecontact: DENIC-1000022-EXAMPLE-ABUSE
Nserver: ns1.denic.de
Nserver: ns2.denic.de
CTID: cba-987654321
```

format: k/v

response: domainCREATE

status: pendingCreate

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

INFO: 53000080014 Domain "Status" is "pendingCreate"
STID: 3727d839-497f-4ef9-8aa5-0a1ea16a8c95
CTID: cba-987654321
```

domainUPDATE Request

Function

This request is used to update the data of a domain.

Requirements

The request can only be processed if the domain exists and is administered by the RegAcc that submits the request.

Features

- You cannot use an UPDATE request to change the domain holder. To change the holder, use the CHHOLDER request.
- If you omit optional keywords, any values entered for these keywords will be deleted and they will be set to their defaults.
- If new name servers are specified for a UPDATE request and the predelegation check fails for this, the request is rejected with an "error" and the owner data is not changed.
- The order of the name servers can change as a result of a domain UPDATE request. The name server order is always output in the same way in the query with the domain INFO request. A new update can change the order again.

Verification

- A domain UPDATE request automatically triggers a risk assessment.
- A domain UPDATE is still possible if contacts without a phone number or email are used.
- When the request is forwarded to the risk assessment, a notification is generated when the new protocol status changes to “pendingCreate” (transition “failed” → “pendingCreate”).

Request Parameters

A Domain UPDATE request is composed of the fields of the "Domain" data object and some other parameters, which are described below:

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Action	domain:up-date	1	enu-meration	update-erule	This is the request type.
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any vis-ible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

- The domain stated in the request does not exist.
- One of the Contacts stated in the request does not meet the requirements of the role to be assigned to it.
- The Holder stated in the request is not identical with the current holder of the domain.

Examples

- xml
- k/v

format: xml

request: domainUPDATE

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:dnsentry="http://registry.denic.de/dnsentry/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <domain:update>
    <domain:handle>domain-example-nsentry.de</domain:handle>
    <domain:contact role="holder">DENIC-1000022-EXAMPLE-XML-PERSON</domain:contact>
    <domain:contact role="abusecontact">DENIC-1000022-EXAMPLE-XML-GR</domain:contact>
    <domain:contact role="generalrequest">DENIC-1000022-EXAMPLE-XML-ABUSE</domain:contact>
    <dnsentry:dnsentry xsi:type="dnsentry:A">
      <dnsentry:owner>domain-example-nsentry.de</dnsentry:owner>
      <dnsentry:rdata>
        <dnsentry:address>81.91.170.12</dnsentry:address>
      </dnsentry:rdata>
    </dnsentry:dnsentry>
  </domain:update>
  <ctid>xml-74ba5156</ctid>
</registry-request>
```

format: xml

response: domainUPDATE

status: pendingCreate

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>a503e644-635c-4253-b598-84d0c2296a97</tr:stid>
    <tr:result>success</tr:result>
    <tr:ctid>xml-74ba5155</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080014">
      <tr:text>Domain "Status" is "pendingCreate"</tr:text>
    </tr:message>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainUPDATE

```
Version: 5.0
Action: UPDATE
Domain: domain-example-nsentry.de
Holder: DENIC-1000022-EXAMPLE-PERSON
Generalrequest: DENIC-1000022-EXAMPLE-GR
Abusecontact: DENIC-1000022-EXAMPLE-ABUSE
Nsentry: domain-example-nsentry.de IN A 127.0.0.1
```

format: k/v

response: domainUPDATE

status: pendingCreate

```
Result: success
STID: afdd1680-6d5e-4e94-a77a-8ef4a267c6b1
Info: 53000080014 Domain "Status" is "pendingCreate"
```

domainCHHOLDER Request

Function

This request is used to change the domain holder(s). You may add new holders and / or remove existing ones.

Requirements

The request can only be processed if the domain exists and is administered by the RegAcc that submits the request.

Features

- When a domain is transferred to a new domain holder within the framework of a CHHOLDER, all AuthInfo entries stored for this domain will become invalid.
- If a domain is subject to an active DISPUTE, the domain holder can only be changed (CHHOLDER) if the mandatory data for the holder remains unaltered.
- If new name servers are specified for a CHHOLDER request and the predelegation check fails for this, the request is rejected with an "error" and the owner data is not changed.

Verification

- A domain CHHOLDER order automatically triggers a risk assessment.
- Domain CHHOLDER is also executed when contacts without/with telephone number or e-mail are to be changed.
- During the process implementation, additional messages may appear for domain CHHOLDER if deadlines are set.

Request Parameters

A CHHOLDER request is composed of the "Domain" data object fields and some other parameters, which are described below:

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Action	domain:ch-holder	1	enu-meration	chholder-erule	This is the request type.
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any vis-ible Unicode character (accord-ing to Unicode version 3.1	This is the unique transaction ID which is issued by the client.

Common Mistakes

The domain stated in the request does not exist.

Examples

- xml
- k/v

format: xml
request: domainCHHOLDER

format: xml**request:** domainCHHOLDER

format: xml

response: domainCHHOLDER

status: connect

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "connect"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
    <tr:message level="info" code="16350000040">
      <tr:text>Verification information must be provided for the holder(s) to avoid dedelegation by</tr:text>
      <tr:argument>2024-12-09T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: k/v

response: domainCHHOLDER

status: serverHold

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
```

format: k/v**response:** domainCHHOLDER**status:** serverHold

```
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "serverHold"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: k/v

request: domainCHHOLDER

```
Version: 5.0
Action: chholder
CTID: cba-9345345321
Domain: de-example.de
Holder: DENIC-1000002-MAX
Generalrequest: DENIC-1000002-GENERAL
Abusecontact: DENIC-1000002-ABUSE
Nserver: ns1.xn--de-xample-x2a.de
Nserver: ns2.de-example.de 81.91.170.12
Nserver: ns2.de-example.de 2001:608:6:6:0:0:0:11
Dnskey: 257 3 8 AwEAA-
coFUSy-
g1mkE5c33q8UbDiRZx5+/QtqFjVcyTEd-
v7YBYp9Un-
qrbXr7g4p8aDMI0ZuN4M8bxlPz+ItVfWO71rSk-
cxK1HwqmH4Pi1vSM3L6uYqZopEG9gJLqNpBBmzR29iSwR86Td-
nGUJ21Jfagc/+9xk3xmtdzNK3ROUcn/f8yiBN
```

format: k/v

response: domainCHHOLDER

status: connect

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

Info: 53000080013 Domain "Status" is "connect"
Info: 16350000040 Verification information must be provided for the holder(s) to avoid dedelegation by [2024-12-09T15:45:01+01:00]
Info: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
STID: 57ebbb10-9590-11df-a9e2-9519b5688882
CTID: cba-9345345321
```

format: k/v

response: domainCHHOLDER

status: serverHold

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

Info: 53000080015 Domain "Status" is "serverHold"
Info: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
STID: 57ebbb10-9590-11df-a9e2-9519b5688882
CTID: cba-9345345321
```

domainRESTORE Request

Function

This request is used to re-activate a domain.

Requirements

The request can only be processed if the status of the domain is „redemptionPeriod“ and if the domain is administered by the RegAcc that submits the request.

If the domain shall be restored by another RegAcc, you must first request an AuthInfo2 for the domain. Afterwards, the domain can be restored by a CHPROV request containing the AuthInfo2.

Features

If prior to it´s deletion the status of the domain has been “failed”, a new expire date of 30 days from the day of the RESTORE will be set.

Verification

- A domain that was deleted due to unverified contact data will not receive a Redemption Grace Period.
- This is to prevent the domain from being restored via a Domain RESTORE request with unverified contact data.
- A Domain RESTORE without an AuthInfo is still possible with contacts without a phone number.

Request Parameters

A RESTORE request is composed of the "Domain" data object fields and some other parameters, which are described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	domain:restore	1	enumeration	restore-erule	This is the request type.
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/-value format.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
CtId	ctid	0 - 1	token 3 - 64	Any vis-ible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.
Domain	domain:handl-e	1	token 4 - 66	domain-rule	Name of the Domain

Common Mistakes

The domain stated in the request is not in the Redemption Grace Period.

Examples

- xml
- k/v

format: xml

request: domainRESTORE

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0"
xmlns:domain="http://registry.denic.de/domain/5.0">
  <domain:restore>
  <domain:handle>de-example.de</domain:handle>
  </domain:restore>
  <ctid>cba-987654321</ctid>
</registry-request>
```

format: xml

response: domainRESTORE

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-98765431</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainRESTORE

```
Version: 5.0
Action: restore
CTID: cba-987654321
Domain: de-example.de
```

format: k/v

response: domainRESTORE

```
Result: success
STID: 1aba2fc0-9625-11df-a9e2-9519b5688882
CTID: cba-987654321
```

domainDELETE Request

Function

This request is used to delete a domain.

Requirements

The request can only be processed if the domain exists and is administered by the RegAcc that submits the request.

Features

- Holder is an optional keyword in a DELETE request. If you do state a holder, the data must be identical with that recorded in the DENIC database. Otherwise the request will be rejected with an error message.
- All other optional data (such as nserver) are ignored.
- The domain will be in a „redemptionPeriod“ status for a certain time.
- When the domain is deleted, an existing AuthInfo1 and AuthInfo2 for the domain is also deleted.
- If the domain which is requested to be deleted bears a DISPUTE entry, it will automatically be re-registered immediately upon deletion. The DISPUTE holder will become the new domain holder.
- If the domain was deleted by DENIC (e.g. due to a final judgement) it is available for registration immediately. There is no RGP (Redemption Grace Period) in this case.
- If a legitimate interest has been proven, DENIC can remove a domain from the „redemptionPeriod“ status. In this case the domain can be registered immediately.

Request Parameters

A DELETE request is composed of the parameters described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	domain:delete	1	enumeration	deleterule	This is the request type.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any vis-ible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.
Domain	domain:handl-e	1	token 4 - 66	domain-rule	Name of the domain
Holder	domain:-contact role-e="holder"	0-5	token 9 - 32	contact-rule	Handle ID of the domain holder

Notice

The final deletion of a domain is executed upon expiration of the Redemption Grace Period (e.g. after 30 days) at an accidental point of time within a defined time interval (e.g. between 2 and 4 a.m.

DENIC reserves the right to modify at any time the term of the Redemption Grace Period and the time interval during which the domain will be finally deleted!

Examples

- xml
- k/v

format: xml

request: domainDELETE

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0"
xmlns:domain="http://registry.denic.de/domain/5.0">
  <domain:delete>
  <domain:handle>de-example.de</domain:handle>
  <domain:contact role="holder">DENIC-1000002-MAX</domain:contact>
  </domain:delete>
  <ctid>cba-987654321</ctid>
</registry-request>
```

format: xml

response: domainDELETE

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-98765431</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainDELETE

```
Version: 5.0
Action: delete
CTID: cba-987654321
Domain: de-example.de
Holder: DENIC-1000002-MAX
```

format: k/v

response: domainDELETE

```
Result: success
STID: 1aba2fc0-9625-11df-a9e2-9519b5688882
CTID: cba-987654321
```

domainTRANSIT Request

Function

This request is used to place a domain in the administration of DENIC (TRANSIT). The RegAcc administering the domain up to that point in time will no longer have access to it.

Requirements

TRANSIT requests can be submitted only for domains of the requestor's own RegAcc. The domain must not have the status "failed".

Features

- When a domain is returned to DENIC, the AuthInfo1 which was stored with DENIC by the administering RegAcc becomes invalid and is deleted. The RegAcc placing the request is informed accordingly in the request acknowledgement. A possibly existing AuthInfo2 is not affected by the TRANSIT request.
- TRANSIT becomes impossible as soon as a domain has the status "failed".
- A TRANSIT request removes an existing General Request / Abuse Contact.

Verification

- The verification period is lifted if the TRANSIT letter could be delivered (the letter does not come back to DENIC and the code in the letter was used).
- The TRANSIT deadlines apply.
- After a move out of TRANSIT, a new risk assessment is carried out as part of the provider change.

Request Parameters

A request comprises the parameters described below:

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Action	domain:trans-it	1	enu-meration	transit-erule	This is the request type.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
Domain	domain:hand-le	1	token 4 - 66	domain-rule	This is the name of the domain.
Dis-connect	"disconnect" attribute for the " domain:trans-it " element	0 - 1	enu-meration	boolean-erule	true = The prior name server entries will be replaced by an NS-Entry that makes ref-erence to a DENIC webpage. false = The previous nameserve-r entries remain.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
CtId	ctid	0 - 1	token 3 - 64	Any vis-ible Unicode character (accord-ing to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

The domain stated in the request does not exist.

One of the keywords stated in the request is invalid (e.g. deconnect instead of dis-connect); the domain remains connected.

Examples

- xml
- k/v

format: xml

request: domainTRANSIT

disconnect: false

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0"
xmlns:domain="http://registry.denic.de/domain/5.0">
  <domain:transit disconnect="false">
  <domain:handle>de-example.de</domain:handle>
  </domain:transit>
  <ctid>cba-987654321</ctid>
</registry-request>
```

format: xml

response: domainTRANSIT

disconnect: false

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>342d1d01-963c-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: xml

request: domainTRANSIT

disconnect: true

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0"
xmlns:domain="http://registry.denic.de/domain/5.0">
  <domain:transit disconnect="true">
  <domain:handle>de-example.de</domain:handle>
  </domain:transit>
  <ctid>cba-987654321</ctid>
</registry-request>
```

format: xml

response: domainTRANSIT

disconnect: true

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>342d1d01-963c-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainTRANSIT

disconnect: false

```
Version: 5.0
Action: transit
CTID: cba-987654321
Domain: de-example.de
Disconnect: false
```

format: k/v

response: domainTRANSIT

disconnect: false

```
Result: success
STID: 1aba2fc0-9625-11df-a9e2-9519b5688882
CTID: cba-987654321
```

format: k/v

request: domainTRANSIT

disconnect: true

```
Version: 5.0
Action: transit
CTID: cba-987654321
Domain: de-example.de
Disconnect: true
```

format: k/v

response: domainTRANSIT

disconnect: true

```
Result: success
STID: 1aba2fc0-9625-11df-a9e2-9519b5688882
CTID: cba-987654321
```

Replies to Domain CREATE, UPDATE, CHHOLDER, RESTORE, DELETE, TRANSIT Requests

Description of Data Fields

Replies of the .de registration system to domain requests of the types CREATE, UPDATE, DELETE, CHHOLDER, RESTORE or TRANSIT comprise the data fields described below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of processing: success = request was executed failed = request was not executed exception = system failure. Request was not executed
StId	stid	1	token 30 - 255	stid-rule	A StId is a Server Transaction ID which is assigned by the server. The StId ensures that the request from the provider and DENIC's reply are referenced with a globally unique ID.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
CtId	ctid	0 - 1	token 3 - 64	Any visible Unicode char-acter (accord-ing to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.
Error	message level-l="error" code=	0 - *	token 1 - 255	message-rule	This is the error code for failed requests. One reply may contain more than one error code.

domainCHECK Request

Function

This request is used to find out about the status of a domain.

Requirements

None.

Request Parameters

Contact CHECK request comprises the parameters described below:

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Action	domain:check	1	enu-meration	check-erule	This is the request type.
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
Domain	domain:hand-le	1	token 4 - 66	domain-rule	This is the domain name.
Domain-Ace	domain:ace	0 - 1	token 4 - 66	domain-ace-rule	This is the ACE form (ASCII Com-patible Encoding) of the domain name.

Examples

- xml
- k/v

format: xml

request: domainCHECK

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0"
xmlns:domain="http://registry.denic.de/domain/5.0">
  <domain:check>
    <domain:handle>de-example.de</domain:handle>
  </domain:check>
</registry-request>
```

format: xml

response: domainCHECK

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>9e986503-9a2c-11df-a9e2-9519b5688882</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
      <domain:checkData xmlns:domain="h-
ttp://registry.denic.de/domain/5.0">
        <domain:handle>test.de</domain:handle>
        <domain:status>connect</domain:status>
      </domain:checkData>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

response: domainCHECK

status: pendigCreate

```
<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>0a38e709-33e3-d5cf-cccf-622f9a16f225</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
      <domain:checkData xmlns:domain="h-
ttp://registry.denic.de/domain/5.0">
        <domain:handle>de-example.de</domain:handle>
        <domain:ace>de-example.de</domain:ace>
        <domain:status>pendingCreate</domain:status>
      </domain:checkData>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

response: domainCHECK

status: serverHold

```
<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>0a38e709-33e3-d5cf-cccf-622f9a16f225</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
      <domain:checkData xmlns:domain="h-
ttp://registry.denic.de/domain/5.0">
        <domain:handle>denic-server-hold.de</domain:handle>
        <domain:ace>denic-server-hold.de</domain:ace>
        <domain:status>serverHold</domain:status>
      </domain:checkData>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainCHECK

```
Version: 5.0
Action: check
Domain: de-example.de
```

format: k/v

response: domainCHECK

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

Domain: de-example.de
Status: exist
```

format: k/v

response: domainCHECK

status: pendigCreate

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

Domain: de-example.de
Domain-Ace: de-example.de
Status: pendigCreate
```

format: k/v

response: domainCHECK

status: serverHold

```
Result: success
```

```
2 | STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d
3 |
4 | Domain: de-example.de
5 | Domain-Ace: de-example.de
6 | Status: serverHold
```

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

domainINFO Request

Function

This request is used to query the data of a domain.

Requirements

The domain must exist.

Features

The data output varies, depending on whether the field "recursive" has the value "true" or "false", see Documentation DENIC-31.

With an Authinfo1 or Authinfo2, the owner data (Holder) of own and foreign domains can be queried as often as required until the expire date or until used in a CHPROV or CHHOLDER request.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	domain:info	1	enumeration	info-erule	This is the request type.
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/value format.
Domain	domain:handle	1	token 4 - 66	domain-rule	This is the name of the domain.
Domain-Ace	domain:ace	0 - 1	token 4 - 66	domainace-rule	This is the ACE form (ASCII Compatible Encoding) of the domain name.
AuthInfo	domain:authInfo	0 -1	enumeration	-	AuthInfo1 or AuthInfo2
Recursive	"recursive" attribute for the "domain:info" element	0 - 1	boolean	boolean-erule	If "recursive = true" is set, the Contact data will be included in the output.

Common Mistakes

The domain stated in the request does not exist.

Examples

For a domain INFO response, the data is grouped into multiple records in either CSV or XML format.

In the domain INFO response, the number of records and their level of detail can vary. This depends on the query parameters and whether the RegAcc used to query the domain information has the required authorization.

Explanation of the examples in the tables

IThe table explains the data records from the left column of the comparison table with the examples (at the end of the topic, can be expanded).

Record	Line Numbers	Explanation
Status query	1 - 2	Confirmation of whether the query was successful or failed
Domain data	4 - 11	• Domain name • Domain name in ace format • Name server (or ns entry) • Status of the domain • Registration date • Deadlines • Date of the data set; is updated when at least one row has been changed
General Request (optional)	13 - 17	• Header • Handle of the contact • Type, indicating the contact type REQUEST • URI template, in raw format, which means that the possible placeholders for “A-label” and “U-label” are not replaced with real data • Date of the data set; is updated when at least one row has been changed
Abuse Contact (optional)	19 - 23	The meaning of the information in the data set for “Abuse Contact” is the same as for the data set “General Request”.
Domain holder	25 - 35	• Header, Domain holder data only if the holder is of type “ORG” • For a domain holder, the information about the address contact is possible as described in the DENIC-29 Contact Data Object. • The telephone number has been added.
Verification informations (optional)	37 - 44	For details, see Verification - Formats, Types, Values

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

Record	Line Numbers	Explanation
Further domain holders and verification informations (optional)	46 - 75	-
Registration account		Registrar information: • Header • RegAcc ID • Name • Email • Date of the data set; is updated when at least one row has been changed

Special feature of the domain owner and the verification information in K/V format

- In the K/V request, the domain holder and verification data records that belong together are listed one below the other.
- The date ("changed”) is displayed for domain holders, but it also applies to the associated verification data.

Special feature of the registration date

- In isolated cases, no registration date may be available. Instead, the earliest internal system process date that can be determined for a domain is displayed. Domains without a registration date are displayed with the prefix “before”:

```
RegistrationDate: before 2011-05-31T00:00:00+02:00
```

Special feature of the "changed" date

In the case of a foreign domain, changed data may not be visible because the information is restricted due to the parameters of the request.

This is the case in the verification information, for example, if the value for “VerificationReference” has been changed.

In the result of the request, the domain owner record (type ORG) displays the changed date, but "VerificationReference" is not visible in the associated verification information.

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

Special note on deadlines

If a risk assessment has been carried out for a domain and the result is that the contact(s) must be verified, the deadline lines will show date and time information indicating a deadline. Verification must be carried out by these deadlines, otherwise certain actions will be taken when the dates expire:

- **VerificationDeadlineBeforeDedelegation:** When the deadline expires, the name servers for the domain will be removed from the.de zone.
- **VerificationDeadlineBeforeDeletion:** Once the deadline has passed, the domain is deleted (DELETE) by DENIC, without RGP.

Deadlines are not issued with an invalid AuthInfo password.

Example

```
VerificationDeadlineBeforeDedelegation: 2024-06-05T00:00:00+02:00
VerificationDeadlineBeforeDeletion: 2024-09-05T00:00:00+02:00
```

The format of the time specification is explained under Verification - Formats, Types, Values.

Special feature of the RegAcc data set

More detailed information about the registration account and the member can be requested with the RegAcc INFO request. A domain INFO request displays the compact information shown in the examples.

Special feature of keywords

In compound words, the Camelcase spelling is used, in which the first letter of a word is capitalized.

Special features of the domain status

The possible statuses are displayed under Domain INFO: “connect”, “failed”, “pendingCreate” and “serverHold”. For more information on “pendingCreate” and “serverHold”, see "domainCREATE Request" on page 324.

> **Caution!**
>
> When making a request using RRI protocol version 4.0, the status “failed” is issued for domains that are in the “pendingCreate” or “serverHolder” status.

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

Response Examples

The table compares the results of Domain INFO responses queried with different administrative statuses and parameters.

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

Examples

1

format: k/v
response: domainINFO
recursive: true
regacc: you
handle-type: PERSON or ORG

```
RESULT: success
STID: 99849ee7-503b-46b4-868d-5ab3efa33a3c

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
Veri-fic-ationDead-lineBeforeDedelegation: 2024-06-05T00:00:00+02:00
Veri-fic-ationDead-lineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]
Handle: DENIC-1000002-DENIC-GR
Type: REQUEST
URI-Template: mailto:db-s@denic.de
Changed: 2018-08-27T15:14:15+02:00

[Abuse Contact]
Handle: DENIC-1000002-DENIC-ABUSE
Type: REQUEST
```

2

format: k/v
response: domainINFO
recursive: false
regacc: you
handle-types: PERSON or ORG

```
RESULT: success
STID: c7c97267-5d28-4362-ba95-9b4ae3109ab7

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
Veri-fic-ationDead-lineBeforeDedelegation: 2024-06-05T00:00:00+02:00
Veri-fic-ationDead-lineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]
Handle: DENIC-1000022-DENIC-GR

[Abuse Contact]
Handle: DENIC-1000022-DENIC-ABUSE

[Holder]
Handle: DENIC-1000022-DENIC-ORG
```

3

format: k/v
response: domainINFO
recursive: true
authinfo: set
regacc: others
handle-types: PERSON or ORG

```
Result: success
STID: 0a38e709-33e3-d5cf-cccf-622f9a16f225

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
Veri-fic-ationDead-lineBeforeDedelegation: 2024-06-05T00:00:00+02:00
Veri-fic-ationDead-lineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]
Type: REQUEST
URI-Template: mailto:db-s@denic.de
Changed: 2024-10-04T13:21:25+02:00

[Abuse Contact]
Type: REQUEST
```

4

format: k/v
response: domainINFO
recursive: false
authinfo: set
regacc: others
handle-types: PERSON or ORG

```
RESULT: success
STID: 0c71c5a3-70eb-4194-b6de-4282ff414b44

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
Veri-fic-ationDead-lineBeforeDedelegation: 2024-06-05T00:00:00+02:00
Veri-fic-ationDead-lineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]

[Abuse Contact]

[Holder]

[RegAcc]
RegAccId: DENIC-1000022
Name: DENIC eG
```

5

format: k/v
response: domainINFO
recursive: true
regacc: others
handle-types: PERSON or ORG

```
RESULT: success
STID: 6c4a38f9-07d2-42d1-8fb0-8124e66ac6ed

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
Veri-fic-ationDead-lineBeforeDedelegation: 2024-06-05T00:00:00+02:00
Veri-fic-ationDead-lineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]
Type: REQUEST
URI-Template: mailto:db-s@denic.de
Changed: 2018-08-27T15:14:15+02:00

[Abuse Contact]
Type: REQUEST
URI-Template: mailto:-abuse@denic.de
Changed: 2018-08-27T15:14:15+02:00
```

6

format: k/v
response: domainINFO
recursive: false
regacc: others
handle-types: PERSON or ORG

```
RESULT: success
STID: 534b0ec4-83fa-4255-9601-b110e47616be

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
Regis-trationDate: 2024-10-04T13:31:25+02:-00
Changed: 2024-10-04T18:23:08+02:-00

[General Request]

[Abuse Contact]

[Holder]

[Holder]

[RegAcc]
RegAccId: DENIC-1000022
Name: DENIC eG
```

1

```
URI-Template: mailto:-abuse@denic.de
Changed: 2018-08-27T15:14:15+02:00

[Holder]
Handle: DENIC-1000002-DENIC-ORG
Type: ORG
Name: DENIC eG
Address: Theodor-Stern-Kai 1-3
City: Frankfurt am Main
PostalCode: 60596
CountryCode: DE
Email: info@denic.de
Phone: +49.6927235x290
Changed: 2018-08-27T15:14:15+02:00

[Veri-ficationInformation]
VerifiedClaim: name
VerificationResult: suc-cess
VerificationReference: ABC123/45GHT1
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: company_register
VerificationMethod: elec-tronic_document
TrustFramework: de_denic

[Holder]
Handle: DENIC-1000002-PERSON
Type: PERSON
Name: Max Mustermann
Address: Musterstraße 1
City: Musterstadt
PostalCode: 99999
CountryCode: DE
Email: max@mustermann.de
Phone: +49.1731234567
Changed: 2018-08-27T15:14:15+02:00
```

2

```
[Holder]
Handle: DENIC-1000022-DENIC-PERSON

[RegAcc]
RegAccId: DENIC-1000022
Name: DENIC eG
```

3

```
URI-Template: mailto:-abuse@denic.de
Changed: 2024-10-04T13:21:25+02:00

[Holder]
Type: ORG
Name: DENIC eG
Address: Theodor-Stern-Kai 1
City: Frankfurt
PostalCode: 60596
CountryCode: DE
Email: info@denic.de
Phone: +49.6927235x290
Changed: 2024-10-04T13:22:04+02:00

[Veri-ficationInformation]
VerifiedClaim: address
VerifiedClaim: name
VerificationResult: suc-cess

[Holder]
Type: PERSON
Name: John Doe
Address: Theodor-Stern-Kai 1
City: Frankfurt
PostalCode: 60596
CountryCode: DE
Email: max@mustermann.de
Phone: +49.69272350x290
Changed: 2024-10-04T19:01:40+02:00

[Veri-ficationInformation]
VerifiedClaim: address
VerificationResult: suc-cess

[Veri-ficationInformation]
VerifiedClaim: name
VerificationResult: suc-cess
```

4

5

format: k/v**response:** domainINFO**recursive:** true**regacc:** others**handle-types:** PERSON or ORG

```

[Holder]
Type: ORG
Name: DENIC eG
Address: Theodor-Stern-Kai 1
City: Frankfurt
PostalCode: 60596
CountryCode: DE
Email: john.doe@denic.de
Phone: +49.6927235x290
Changed: 2024-10-04T13:22:04+02:00

[Veri-ficationInformation]
VerifiedClaim: name
VerificationResult: suc-cess

[Holder]
Type: PERSON
Name: Max Mustermann
Address: Musterstraße 1
City: Musterstadt
PostalCode: 99999
CountryCode: DE
Email: max@mustermann.de
Phone: +49.1731234567
Changed: 2018-08-27T15:14:15+02:00

[Veri-ficationInformation]
VerifiedClaim: address
VerifiedClaim: name
VerificationResult: suc-cess

[Veri-ficationInformation]
VerifiedClaim: email
```

6

1

```
57
58  [Veri-
    ficationInformation]
59  VerifiedClaim: address
60  VerifiedClaim: name
61  VerificationResult: suc-
    cess
62  VerificationReference:
    ABC123/45GHT2
63  VerificationTimestamp:
    2023-11-
    11T15:36:21+02:00
64  VerificationEvidence:
    idcard
65  VerificationMethod: vdig
66  TrustFramework: de_denic
67
68  [VerificationInformation-
    ]
69  VerifiedClaim: email
70  VerificationResult:
    failed
71  VerificationReference:
    ABC123/45GHT3
72  VerificationTimestamp:
    2023-11-
    11T15:36:21+02:00
73  VerificationEvidence:
    email_ver_transaction_
    log
74  VerificationMethod:
    reachability
75  TrustFramework: de_denic
76
77  [RegAcc]
78  RegAccId: DENIC-22123
79  Name: DENIC Services
    GmbH &amp;amp; Co. KG
80  Email: info@denic-ser-
    vices.de
81  Phone: +49.61516290940
82  Changed: 2018-08-
    27T15:14:15+02:00
```

2

3

```
58  [Veri-
    ficationInformation]
59  VerifiedClaim: email
60  VerificationResult:
    failed
61
62  [RegAcc]
63  RegAccId: DENIC-1000022
64  Name: DENIC eG
65  Email: info@denic.de
66  Phone: +49.6927235272
67  Changed: 2023-09-
    11T14:15:39+02:00
```

4

5

format: k/v**response:** domainINFO**recursive:** true**regacc:** others**handle-types:** PERSON or ORG

```
56  VerificationResult:
    failed
57
58  [RegAcc]
59  RegAccId: DENIC-22123
60  Name: DENIC Services
    GmbH &amp;amp; Co. KG
61  Email: info@denic-ser-
    vices.de
62  Phone: +49.61516290940
63  Changed: 2018-08-
    27T15:14:15+02:00
```

6

- xml
- k/v

format: xml

request:

format: xml**request:** domainINFO**recursive:** false
recursive: false

format: xml
response: domainINFO
recursive: false
regacc: others
handle-types: PERSON or ORG

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>8a62923d-89f9-4fc7-9852-2bec74b5a1f9</tr:stid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="13000000011">
      <tr:text>Request was processed in test environment - not valid in real world</tr:text>
      <tr:argument>testing platform</tr:argument>
    </tr:message>
    <tr:data>
      <domain:infoData xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:dnsentry="http://registry.denic.de/dnsentry/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <domain:handle>domain-example-1000022.de</domain:handle>
        <domain:ace>domain-example-1000022.de</domain:ace>
        <domain:status>pendingCreate</domain:status>
        <domain:contact role="generalrequest"/>
        <domain:contact role="abusecontact"/>
        <domain:contact role="holder"/>
        <dnsentry:dnsentry xsi:type="dnsentry:A">
          <dnsentry:owner>domain-example-1000022.de.</dnsentry:owner>
          <dnsentry:rdata>
            <dnsentry:address>127.0.0.1</dnsentry:address>
          </dnsentry:rdata>
        </dnsentry:dnsentry>
        <domain:registrationDate>2024-10-04T13:31:25+02:00</domain:registrationDate>
        <domain:changed>2024-10-04T18:23:08+02:00</domain:changed>
        <domain:regAcc>
          <domain:regAccId>DENIC-1000022</domain:regAccId>
          <domain:name>DENIC eG</domain:name>
        </domain:regAcc>
      </domain:infoData>
    </tr:data>
```

format: xml**response:** domainINFO**recursive:** false**regacc:** others**handle-types:** PERSON or ORG

```
33    </tr:transaction>
34  </registry-response>
```

format: xml

response: domainINFO

recursive: false

regacc: you

handle-types: PERSON or ORG

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
4    <tr:transaction>
5      <tr:stid>e742d307-dd8b-41dc-8fc1-3db0ff50a5b9</tr:stid>
6      <tr:result>success</tr:result>
7      <tr:message level="info" code="13000000011">
8        <tr:text>Request was processed in test environment - not valid in real world</tr:text>
9        <tr:argument>testing platform</tr:argument>
10     </tr:message>
11     <tr:data>
12       <domain:infoData xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:dnsentry="h-
   ttp://registry.denic.de/dnsentry/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
13         <domain:handle>domain-example-1000022.de</domain:handle>
14         <domain:ace>domain-example-1000022.de</domain:ace>
15         <domain:status>pendingCreate</domain:status>
16         <domain:contact role="generalrequest">
17           <contact:handle>DENIC-1000022-EXAMPLE-GR</contact:handle>
18         </domain:contact>
19         <domain:contact role="abusecontact">
20           <contact:handle>DENIC-1000022-EXAMPLE-ABUSE</contact:handle>
21         </domain:contact>
22         <domain:contact role="holder">
23           <contact:handle>DENIC-1000022-EXAMPLE-ORG</contact:handle>
24         </domain:contact>
25         <domain:contact role="holder">
26           <contact:handle>DENIC-1000022-EXAMPLE-PERSON</contact:handle>
27         </domain:contact>
28         <dnsentry:dnsentry xsi:type="dnsentry:A">
29           <dnsentry:owner>domain-example-1000022.de.</dnsentry:owner>
30           <dnsentry:rdata>
```

format: xml**response:** domainINFO**recursive:** false**regacc:** you**handle-types:** PERSON or ORG

```
            <dnsentry:address>127.0.0.1</dnsentry:address>
          </dnsentry:rdata>
        </dnsentry:dnsentry>
        <domain:registrationDate>2024-10-04T13:31:25+02:00</domain:registrationDate>
        <domain:changed>2024-10-04T18:23:08+02:00</domain:changed>
        <domain:regAcc>
          <domain:regAccId>DENIC-1000022</domain:regAccId>
          <domain:name>DENIC eG</domain:name>
        </domain:regAcc>
      </domain:infoData>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

request: domainINFO

recursive: false

authinfo: set

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:domain="http://registry.denic.de/domain/5.0">
   <domain:info recursive="false">
      <domain:handle>domain-example-1000022.de</domain:handle>
      <domain:authInfo>My-Very-Secret-Auth-Info</domain:authInfo>
   </domain:info>
</registry-request>
```

format: xml

response: domainINFO

recursive: false

authinfo: set

regacc: others

handle-types: PERSON or ORG

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>1f33a361-e9c0-4776-a59f-167305200898</tr:stid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="13000000011">
      <tr:text>Request was processed in test environment - not valid in real world</tr:text>
      <tr:argument>testing platform</tr:argument>
    </tr:message>
    <tr:data>
      <domain:infoData xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:dnsentry="http://registry.denic.de/dnsentry/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <domain:handle>domain-example-1000022.de</domain:handle>
        <domain:ace>domain-example-1000022.de</domain:ace>
        <domain:status>pendingCreate</domain:status>
        <domain:authInfo>true</domain:authInfo>
        <domain:contact role="generalrequest"/>
        <domain:contact role="abusecontact"/>
        <domain:contact role="holder"/>
        <dnsentry:dnsentry xsi:type="dnsentry:A">
          <dnsentry:owner>domain-example-1000022.de.</dnsentry:owner>
          <dnsentry:rdata>
            <dnsentry:address>127.0.0.1</dnsentry:address>
          </dnsentry:rdata>
        </dnsentry:dnsentry>
        <domain:registrationDate>2024-10-04T13:31:25+02:00</domain:registrationDate>
        <domain:changed>2024-10-04T18:23:08+02:00</domain:changed>
        <domain:regAcc>
          <domain:regAccId>DENIC-1000022</domain:regAccId>
          <domain:name>DENIC eG</domain:name>
        </domain:regAcc>
      </domain:infoData>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

request: domainINFO

recursive: true

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:domain="http://registry.denic.de/domain/5.0">
   <domain:info recursive="true">
      <domain:handle>domain-example-1000022.de</domain:handle>
   </domain:info>
</registry-request>
```

format: xml

response: domainINFO

recursive: true

regacc: others

handle-types: PERSON or ORG

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>d0dccb85-2d39-4ebb-8f7c-92f55f1de7b1</tr:stid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="13000000011">
      <tr:text>Request was processed in test environment - not valid in real world</tr:text>
      <tr:argument>testing platform</tr:argument>
    </tr:message>
    <tr:data>
      <domain:infoData xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:dnsentry="h-
ttp://registry.denic.de/dnsentry/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <domain:handle>domain-example-1000022.de</domain:handle>
        <domain:ace>domain-example-1000022.de</domain:ace>
        <domain:status>pendingCreate</domain:status>
        <domain:contact role="generalrequest">
          <contact:type>REQUEST</contact:type>
          <contact:uri-template>mailto:://noreply@denic.de</contact:uri-template>
          <contact:changed>2024-10-04T13:21:16+02:00</contact:changed>
        </domain:contact>
        <domain:contact role="abusecontact">
          <contact:type>REQUEST</contact:type>
```

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

format: xml**response:** domainINFO**recursive:** true**regacc:** others**handle-types:** PERSON or ORG

```
            <contact:uri-template>mailto::://noreply@denic.de</contact:uri-template>
            <contact:changed>2024-10-04T13:21:25+02:00</contact:changed>
          </domain:contact>
          <domain:contact role="holder">
            <contact:type>ORG</contact:type>
            <contact:name>DENIC eG</contact:name>
            <contact:postal>
              <contact:address>Theodor-Stern-Kai 1</contact:address>
              <contact:postalCode>60596</contact:postalCode>
              <contact:city>Frankfurt</contact:city>
              <contact:countryCode>DE</contact:countryCode>
            </contact:postal>
            <contact:email>john.doe@denic.de</contact:email>
            <contact:phone>+49.6927235x290</contact:phone>
            <contact:changed>2024-10-04T13:22:04+02:00</contact:changed>
          </domain:contact>
          <domain:contact role="holder">
            <contact:type>PERSON</contact:type>
          </domain:contact>
          <dnsentry:dnsentry xsi:type="dnsentry:A">
            <dnsentry:owner>domain-example-1000022.de.</dnsentry:owner>
            <dnsentry:rdata>
              <dnsentry:address>127.0.0.1</dnsentry:address>
            </dnsentry:rdata>
          </dnsentry:dnsentry>
          <domain:registrationDate>2024-10-04T13:31:25+02:00</domain:registrationDate>
          <domain:changed>2024-10-04T18:23:08+02:00</domain:changed>
          <domain:regAcc>
            <domain:regAccId>DENIC-1000022</domain:regAccId>
            <domain:name>DENIC eG</domain:name>
            <domain:email>1000022@opsblau.de</domain:email>
            <domain:phone>+49 69 27235 272</domain:phone>
            <domain:changed>2023-09-11T14:15:39+02:00</domain:changed>
          </domain:regAcc>
        </domain:infoData>
      </tr:data>
    </tr:transaction>
</registry-response>
```

format: xml

response: domainINFO

recursive: true

regacc: you

handle-type: PERSON or ORG

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>9ca20332-cc31-436e-9f05-6160cd81845a</tr:stid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="13000000011">
      <tr:text>Request was processed in test environment - not valid in real world</tr:text>
      <tr:argument>testing platform</tr:argument>
    </tr:message>
    <tr:data>
      <domain:infoData xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:dnsentry="http://registry.denic.de/dnsentry/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <domain:handle>domain-example-1000022.de</domain:handle>
        <domain:ace>domain-example-1000022.de</domain:ace>
        <domain:status>pendingCreate</domain:status>
        <domain:contact role="generalrequest">
          <contact:handle>DENIC-1000022-EXAMPLE-GR</contact:handle>
          <contact:type>REQUEST</contact:type>
          <contact:uri-template>mailto:://noreply@denic.de</contact:uri-template>
          <contact:changed>2024-10-04T13:21:16+02:00</contact:changed>
        </domain:contact>
        <domain:contact role="abusecontact">
          <contact:handle>DENIC-1000022-EXAMPLE-ABUSE</contact:handle>
          <contact:type>REQUEST</contact:type>
          <contact:uri-template>mailto:://noreply@denic.de</contact:uri-template>
          <contact:changed>2024-10-04T13:21:25+02:00</contact:changed>
        </domain:contact>
        <domain:contact role="holder">
          <contact:handle>DENIC-1000022-EXAMPLE-ORG</contact:handle>
          <contact:type>ORG</contact:type>
          <contact:name>DENIC eG</contact:name>
          <contact:postal>
            <contact:address>Theodor-Stern-Kai 1</contact:address>
            <contact:postalCode>60596</contact:postalCode>
            <contact:city>Frankfurt</contact:city>
            <contact:countryCode>DE</contact:countryCode>
          </contact:postal>
          <contact:email>john.doe@denic.de</contact:email>
          <contact:phone>+49.6927235x290</contact:phone>
```

format: xml**response:** domainINFO**recursive:** true**regacc:** you**handle-type:** PERSON or ORG

```
          <contact:changed>2024-10-04T13:22:04+02:00</contact:changed>
          <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:type="ver-
rification:domainVerificationInformationType">
            <verification:verifiedClaims>
              <verification:claim>address</verification:claim>
              <verification:claim>name</verification:claim>
            </verification:verifiedClaims>
            <verification:verificationResult>success</verification:verificationResult>
            <verification:verificationReference>ABC123/45GHT</verification:verificationReference>
            <verification:verificationTimestamp>2023-11-11T10:11:00+01:00</verification:verificationTimestamp>
            <verification:verificationEvidence>company_register</verification:verificationEvidence>
            <verification:verificationMethod>electronic_document</verification:verificationMethod>
            <verification:trustFramework>de_denic</verification:trustFramework>
          </verification:verificationInformation>
        </domain:contact>
        <domain:contact role="holder">
          <contact:handle>DENIC-1000022-EXAMPLE-PERSON</contact:handle>
          <contact:type>PERSON</contact:type>
          <contact:name>John Doe</contact:name>
          <contact:postal>
            <contact:address>Theodor-Stern-Kai 1</contact:address>
            <contact:postalCode>60596</contact:postalCode>
            <contact:city>Frankfurt</contact:city>
            <contact:countryCode>DE</contact:countryCode>
          </contact:postal>
          <contact:email>john.doe@denic.de</contact:email>
          <contact:phone>+49.69272350</contact:phone>
          <contact:changed>2024-10-04T19:01:40+02:00</contact:changed>
          <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:type="ver-
rification:domainVerificationInformationType">
            <verification:verifiedClaims>
              <verification:claim>address</verification:claim>
            </verification:verifiedClaims>
            <verification:verificationResult>success</verification:verificationResult>
            <verification:verificationReference>ABC123/45GHA</verification:verificationReference>
            <verification:verificationTimestamp>2023-11-11T10:11:00+01:00</verification:verificationTimestamp>
            <verification:verificationEvidence>idcard</verification:verificationEvidence>
            <verification:verificationMethod>auth</verification:verificationMethod>
            <verification:trustFramework>de_denic</verification:trustFramework>
          </verification:verificationInformation>
          <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:type="ver-
rification:domainVerificationInformationType">
            <verification:verifiedClaims>
              <verification:claim>name</verification:claim>
            </verification:verifiedClaims>
            <verification:verificationResult>success</verification:verificationResult>
            <verification:verificationReference>ABC123/45GHN</verification:verificationReference>
```

format: xml**response:** domainINFO**recursive:** true**regacc:** you**handle-type:** PERSON or ORG

```
            <verification:verificationTimestamp>2023-11-11T10:11:00+01:00</verification:verificationTimestamp>
            <verification:verificationEvidence>idcard</verification:verificationEvidence>
            <verification:verificationMethod>auth</verification:verificationMethod>
            <verification:trustFramework>de_denic</verification:trustFramework>
          </verification:verificationInformation>
        </domain:contact>
        <dnsentry:dnsentry xsi:type="dnsentry:A">
          <dnsentry:owner>domain-example-1000022.de.</dnsentry:owner>
          <dnsentry:rdata>
            <dnsentry:address>127.0.0.1</dnsentry:address>
          </dnsentry:rdata>
        </dnsentry:dnsentry>
        <domain:registrationDate>2024-10-04T13:31:25+02:00</domain:registrationDate>
        <domain:changed>2024-10-04T18:23:08+02:00</domain:changed>
        <domain:regAcc>
          <domain:regAccId>DENIC-1000022</domain:regAccId>
          <domain:name>DENIC eG</domain:name>
          <domain:email>1000022@opsblau.de</domain:email>
          <domain:phone>+49 69 27235 272</domain:phone>
          <domain:changed>2023-09-11T14:15:39+02:00</domain:changed>
        </domain:regAcc>
      </domain:infoData>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

request: domainINFO

recursive: true

authinfo: set

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:domain="http://registry.denic.de/domain/5.0">
   <domain:info recursive="true">
      <domain:handle>domain-example-1000022.de</domain:handle>
      <domain:authInfo>My-Very-Secret-Auth-Info</domain:authInfo>
   </domain:info>
</registry-request>
```

format: xml

response: domainINFO

recursive: true

authinfo: set

regacc: others

handle-types: PERSON or ORG

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>90ffa070-cd13-4f19-8868-d8a29a32fa84</tr:stid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="13000000011">
      <tr:text>Request was processed in test environment - not valid in real world</tr:text>
      <tr:argument>testing platform</tr:argument>
    </tr:message>
    <tr:data>
      <domain:infoData xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:contact="http://registry.denic.de/contact/5.0" xmlns:dnsentry="http://registry.denic.de/dnsentry/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <domain:handle>domain-example-1000022.de</domain:handle>
        <domain:ace>domain-example-1000022.de</domain:ace>
        <domain:status>pendingCreate</domain:status>
        <domain:authInfo>true</domain:authInfo>
        <domain:contact role="generalrequest">
          <contact:type>REQUEST</contact:type>
          <contact:uri-template>mailto::://noreply@denic.de</contact:uri-template>
          <contact:changed>2024-10-04T13:21:16+02:00</contact:changed>
        </domain:contact>
        <domain:contact role="abusecontact">
          <contact:type>REQUEST</contact:type>
          <contact:uri-template>mailto::://noreply@denic.de</contact:uri-template>
          <contact:changed>2024-10-04T13:21:25+02:00</contact:changed>
        </domain:contact>
        <domain:contact role="holder">
          <contact:type>ORG</contact:type>
          <contact:name>DENIC eG</contact:name>
          <contact:postal>
            <contact:address>Theodor-Stern-Kai 1</contact:address>
            <contact:postalCode>60596</contact:postalCode>
            <contact:city>Frankfurt</contact:city>
            <contact:countryCode>DE</contact:countryCode>
          </contact:postal>
          <contact:email>john.doe@denic.de</contact:email>
          <contact:phone>+49.6927235x290</contact:phone>
          <contact:changed>2024-10-04T13:22:04+02:00</contact:changed>
```

format: xml**response:** domainINFO**recursive:** true**authinfo:** set**regacc:** others**handle-types:** PERSON or ORG

```
39          <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:type="ver-
rification:domainVerificationInformationType">
40            <verification:verifiedClaims>
41              <verification:claim>address</verification:claim>
42              <verification:claim>name</verification:claim>
43            </verification:verifiedClaims>
44            <verification:verificationResult>success</verification:verificationResult>
45          </verification:verificationInformation>
46        </domain:contact>
47        <domain:contact role="holder">
48          <contact:type>PERSON</contact:type>
49          <contact:name>John Doe</contact:name>
50          <contact:postal>
51            <contact:address>Theodor-Stern-Kai 1</contact:address>
52            <contact:postalCode>60596</contact:postalCode>
53            <contact:city>Frankfurt</contact:city>
54            <contact:countryCode>DE</contact:countryCode>
55          </contact:postal>
56          <contact:email>john.doe@denic.de</contact:email>
57          <contact:phone>+49.69272350</contact:phone>
58          <contact:changed>2024-10-04T19:01:40+02:00</contact:changed>
59          <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:type="ver-
rification:domainVerificationInformationType">
60            <verification:verifiedClaims>
61              <verification:claim>address</verification:claim>
62            </verification:verifiedClaims>
63            <verification:verificationResult>success</verification:verificationResult>
64          </verification:verificationInformation>
65          <verification:verificationInformation xmlns:verification="http://registry.denic.de/verification/5.0" xsi:type="ver-
rification:domainVerificationInformationType">
66            <verification:verifiedClaims>
67              <verification:claim>name</verification:claim>
68            </verification:verifiedClaims>
69            <verification:verificationResult>success</verification:verificationResult>
70          </verification:verificationInformation>
71        </domain:contact>
72        <dnsentry:dnsentry xsi:type="dnsentry:A">
73          <dnsentry:owner>domain-example-1000022.de.</dnsentry:owner>
74          <dnsentry:rdata>
75            <dnsentry:address>127.0.0.1</dnsentry:address>
76          </dnsentry:rdata>
77        </dnsentry:dnsentry>
78        <domain:registrationDate>2024-10-04T13:31:25+02:00</domain:registrationDate>
79        <domain:changed>2024-10-04T18:23:08+02:00</domain:changed>
80        <domain:regAcc>
81          <domain:regAccId>DENIC-1000022</domain:regAccId>
82          <domain:name>DENIC eG</domain:name>
```

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

format: xml**response:** domainINFO**recursive:** true**authinfo:** set**regacc:** others**handle-types:** PERSON or ORG

```
            <domain:email>1000022@opsblau.de</domain:email>
            <domain:phone>+49 69 27235 272</domain:phone>
            <domain:changed>2023-09-11T14:15:39+02:00</domain:changed>
          </domain:regAcc>
        </domain:infoData>
      </tr:data>
    </tr:transaction>
</registry-response>
```

format: k/v

request: domainINFO

recursive: false

```
Version: 5.0
Action: INFO
Domain: domain-example-1000022.de
Recursive: false
```

format: k/v

response: domainINFO

recursive: false

regacc: others

handle-types: PERSON or ORG

```
RESULT: success
STID: 534b0ec4-83fa-4255-9601-b110e47616be

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 2024-10-04T13:31:25+02:00
Changed: 2024-10-04T18:23:08+02:00

[General Request]

```

```
[Abuse Contact]

[Holder]

[Holder]

[RegAcc]
RegAccId: DENIC-1000022
Name: DENIC eG
```

format: k/v
response: domainINFO
recursive: false
regacc: you
handle-types: PERSON or ORG

```
RESULT: success
STID: c7c97267-5d28-4362-ba95-9b4ae3109ab7

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
VerificationDeadlineBeforeDedelegation: 2024-06-05T00:00:00+02:00
VerificationDeadlineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]
Handle: DENIC-1000022-DENIC-GR

[Abuse Contact]
Handle: DENIC-1000022-DENIC-ABUSE

[Holder]
Handle: DENIC-1000022-DENIC-ORG

[Holder]
Handle: DENIC-1000022-DENIC-PERSON

[RegAcc]
RegAccId: DENIC-1000022
```

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

```
27 | Name: DENIC eG
```

format: k/v

request: domainINFO

recursive: false

authinfo: set

```
1 | Version: 5.0
2 | Action: INFO
3 | Domain: domain-example-1000022.de
4 | Authinfo: My-Very-Secret-Auth-Info
5 | Recursive: false
```

format: k/v

response: domainINFO

recursive: false

authinfo: set

regacc: others

handle-types: PERSON or ORG

```
 1 | RESULT: success
 2 | STID: 0c71c5a3-70eb-4194-b6de-4282ff414b44
 3 |
 4 | Domain: de-example.de
 5 | Domain-Ace: de-example.de
 6 | Nsentry: de-example.de. IN A 81.91.170.12
 7 | Status: connect
 8 | RegistrationDate: 1992-01-01T08:46:32+02:00
 9 | VerificationDeadlineBeforeDedelegation: 2024-06-05T00:00:00+02:00
10 | VerificationDeadlineBeforeDeletion: 2024-09-05T00:00:00+02:00
11 | Changed: 2019-04-05T09:55:37+02:00
12 |
13 | [General Request]
14 |
15 | [Abuse Contact]
```

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

format: k/v**response:** domainINFO**recursive:** false**authinfo:** set**regacc:** others**handle-types:** PERSON or ORG

```
16
17 [Holder]
18
19 [RegAcc]
20 RegAccId: DENIC-1000022
21 Name: DENIC eG
```

format: k/v

request: domainINFO

recursive: true

```
1 Version: 5.0
2 Action: INFO
3 Domain: domain-example-1000022.de
```

format: k/v

response: domainINFO

recursive: true

regacc: others

handle-types: PERSON or ORG

```
 1 RESULT: success
 2 STID: 6c4a38f9-07d2-42d1-8fb0-8124e66ac6ed
 3
 4 Domain: de-example.de
 5 Domain-Ace: de-example.de
 6 Nsentry: de-example.de. IN A 81.91.170.12
 7 Status: connect
 8 RegistrationDate: 1992-01-01T08:46:32+02:00
 9 VerificationDeadlineBeforeDedelegation: 2024-06-05T00:00:00+02:00
10 VerificationDeadlineBeforeDeletion: 2024-09-05T00:00:00+02:00
11 Changed: 2019-04-05T09:55:37+02:00
12
13 [General Request]
14 Type: REQUEST
```

format: k/v**response:** domainINFO**recursive:** true**regacc:** others**handle-types:** PERSON or ORG

```
URI-Template: mailto:dbs@denic.de
Changed: 2018-08-27T15:14:15+02:00

[Abuse Contact]
Type: REQUEST
URI-Template: mailto:abuse@denic.de
Changed: 2018-08-27T15:14:15+02:00

[Holder]
Type: ORG
Name: DENIC eG
Address: Theodor-Stern-Kai 1
City: Frankfurt
PostalCode: 60596
CountryCode: DE
Email: john.doe@denic.de
Phone: +49.6927235x290
Changed: 2024-10-04T13:22:04+02:00

[VerificationInformation]
VerifiedClaim: name
VerificationResult: success

[Holder]
Type: PERSON
Name: Max Mustermann
Address: Musterstraße 1
City: Musterstadt
PostalCode: 99999
CountryCode: DE
Email: max@mustermann.de
Phone: +49.1731234567
Changed: 2018-08-27T15:14:15+02:00

[VerificationInformation]
VerifiedClaim: address
VerifiedClaim: name
VerificationResult: success

[VerificationInformation]
VerifiedClaim: email
VerificationResult: failed

[RegAcc]
RegAccId: DENIC-22123
Name: DENIC Services GmbH &amp;amp; Co. KG
Email: info@denic-services.de
```

format: k/v**response:** domainINFO**recursive:** true**regacc:** others**handle-types:** PERSON or ORG

```
Phone: +49.61516290940
Changed: 2018-08-27T15:14:15+02:00
```

format: k/v

response: domainINFO

recursive: true

regacc: you

handle-type: PERSON or ORG

```
RESULT: success
STID: 99849ee7-503b-46b4-868d-5ab3efa33a3c

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
VerificationDeadlineBeforeDedelegation: 2024-06-05T00:00:00+02:00
VerificationDeadlineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]
Handle: DENIC-1000002-DENIC-GR
Type: REQUEST
URI-Template: mailto:dbs@denic.de
Changed: 2018-08-27T15:14:15+02:00

[Abuse Contact]
Handle: DENIC-1000002-DENIC-ABUSE
Type: REQUEST
URI-Template: mailto:abuse@denic.de
Changed: 2018-08-27T15:14:15+02:00

[Holder]
Handle: DENIC-1000002-DENIC-ORG
Type: ORG
Name: DENIC eG
Address: Theodor-Stern-Kai 1-3
City: Frankfurt am Main
PostalCode: 60596
```

```
CountryCode: DE
Email: info@denic.de
Phone: +49.6927235x290
Changed: 2018-08-27T15:14:15+02:00

[VerificationInformation]
VerifiedClaim: name
VerificationResult: success
VerificationReference: ABC123/45GHT1
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: company_register
VerificationMethod: electronic_document
TrustFramework: de_denic

[Holder]
Handle: DENIC-1000002-PERSON
Type: PERSON
Name: Max Mustermann
Address: Musterstraße 1
City: Musterstadt
PostalCode: 99999
CountryCode: DE
Email: max@mustermann.de
Phone: +49.1731234567
Changed: 2018-08-27T15:14:15+02:00

[VerificationInformation]
VerifiedClaim: address
VerifiedClaim: name
VerificationResult: success
VerificationReference: ABC123/45GHT2
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: idcard
VerificationMethod: vdig
TrustFramework: de_denic

[VerificationInformation]
VerifiedClaim: email
VerificationResult: failed
VerificationReference: ABC123/45GHT3
VerificationTimestamp: 2023-11-11T15:36:21+02:00
VerificationEvidence: email_ver_transaction_log
VerificationMethod: reachability
TrustFramework: de_denic

[RegAcc]
RegAccId: DENIC-22123
Name: DENIC Services GmbH &amp;amp; Co. KG
```

```
Email: info@denic-services.de
Phone: +49.61516290940
Changed: 2018-08-27T15:14:15+02:00
```

format: k/v

request: domainINFO

recursive: true

authinfo: set

```
Version: 5.0
Action: INFO
Domain: domain-example-1000022.de
Authinfo: My-Very-Secret-Auth-Info
```

format: k/v

response: domainINFO

recursive: true

authinfo: set

regacc: others

handle-types: PERSON or ORG

```
Result: success
STID: 0a38e709-33e3-d5cf-cccf-622f9a16f225

Domain: de-example.de
Domain-Ace: de-example.de
Nsentry: de-example.de. IN A 81.91.170.12
Status: connect
RegistrationDate: 1992-01-01T08:46:32+02:00
VerificationDeadlineBeforeDedelegation: 2024-06-05T00:00:00+02:00
VerificationDeadlineBeforeDeletion: 2024-09-05T00:00:00+02:00
Changed: 2019-04-05T09:55:37+02:00

[General Request]
Type: REQUEST
```

```
URI-Template: mailto:dbs@denic.de
Changed: 2024-10-04T13:21:25+02:00

[Abuse Contact]
Type: REQUEST
URI-Template: mailto:abuse@denic.de
Changed: 2024-10-04T13:21:25+02:00

[Holder]
Type: ORG
Name: DENIC eG
Address: Theodor-Stern-Kai 1
City: Frankfurt
PostalCode: 60596
CountryCode: DE
Email: info@denic.de
Phone: +49.6927235x290
Changed: 2024-10-04T13:22:04+02:00

[VerificationInformation]
VerifiedClaim: address
VerifiedClaim: name
VerificationResult: success

[Holder]
Type: PERSON
Name: John Doe
Address: Theodor-Stern-Kai 1
City: Frankfurt
PostalCode: 60596
CountryCode: DE
Email: max@mustermann.de
Phone: +49.69272350x290
Changed: 2024-10-04T19:01:40+02:00

[VerificationInformation]
VerifiedClaim: address
VerificationResult: success

[VerificationInformation]
VerifiedClaim: name
VerificationResult: success

[VerificationInformation]
VerifiedClaim: email
VerificationResult: failed

[RegAcc]
```

Hint Be careful when printing / size is DIN A3.**Tipp** Achte beim Ausdrucken darauf, dass das Format DIN A3 ist.

```
RegAccId: DENIC-1000022
Name: DENIC eG
Email: info@denic.de
Phone: +49.6927235272
Changed: 2023-09-11T14:15:39+02:00
```

Replies to domainCHECK and domainINFO Requests

Important Information

Existing General Request and Abuse Contact information at the domain level is displayed in a Domain INFO request.

If this information does not exist at the domain level, no fallback will be displayed.

For a domain INFO request, the RegAccId and RegAccName are always displayed for foreign and own domains.

Description of Data Fields

The replies of the .de registration system to CHECK or INFO requests comprise the data fields of the "Domain" and "Contact" data objects and of some other parameters described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of processing: success = request was executed failed = request was not executed exception = system failure, request was not executed

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
StId	Stid	1	token 30 - 255	stid-rule	A StId is a Server Transaction ID which is assigned by the server. The StId ensures that the request from the provider and DENIC's reply are referenced with a globally unique ID.
Error	message level="error" code=	0 - *	normalizedString 1 - 255	message-rule	This the error message related to the request. It consists of the message code and text.
Domain	domain:handle	1	token 4 - 66	domain-rule	This is the domain name.
Domain-Ace	domain:ace	0 - 1	token 4 - 66	domainace-rule	This is the ACE form (ASCII Compatible Encoding) of the domain name.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Status	domain:status	1	enumeration	status-erule	This is the status of a domain: free = not registered connect = registered and connected failed = registered but not connected invalid = invalid domain redemptionPeriod = domain in the redemption grace period
RegAccId	domain:regAccId	1	token 7 - 13	regaccid-rule	The RegAccId is issued by DENIC and is composed of "DENIC-" + numerical value. It is used to uniquely identify the RegAcc.
RegAccName	domain:regAccName	1	normalizedString 1 - 255	name-rule	Name des Inhabers des Registrierungs-Accounts.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
AuthInfo1	domain:authInfo1	0 - 1	enumeration	authinfo-rule	For domains in your administration: Date of AuthInfo1 Expire
AuthInfo2	domain:authInfo2	0 - 1	enumeration	authinfo-rule	For domains in your administration: Date of AuthInfo2 Expire
AuthInfo	domain:authInfo	0 - 1	enumeration	authinfo-erule	"true", if an AuthInfo1 or AuthInfo2 exists for a domain not in your administration
Changed	changed	1	dateTime	date-rule	This is the time stamp of the most recent change in the record concerned.
-	domain:disputeData empty element	0 - 1	-	-	Is marked in K/V format by the heading "[Dispute]". Displayed when querying your own domain, that has a dispute entry.

Error Messages Related to Domain Requests

Potential Error Messages

Below you will find a list of error messages that may occur only in relation with the processing of domain requests.

For general error messages which also occur with other types of requests (e.g. for contacts or message queues), refer to the Annex.

Comments on the Table

- The column "Output Format" informs whether the error message may occur with the key/value (= K/V) and / or with the XML format (= XML).
- The column "Request" informs whether the error message will occur when creating (= C) and / or updating (= U) a domain, in relation with a Chholder (=CH), Restore (=R), Delete (=D), Transit (=T), INFO (=DI) or Check (=DC) request.

Error Code	Error Message	Output Format	Request	Brief Description
13350000001	Domain bears a DISPUTE entry and can therefore not be restored	K/V, XML	D	The domain bears a DISPUTE entry that will become effective immediately upon deletion of the domain. Thus, the Redemption Grace Period is not applicable to the domain.

Error Code	Error Message	Output Format	Request	Brief Description
16000063119	DNS data was checked and accepted	K/V, XML	C, U, CH	INFO message: The Nameserver Predelegation Check was successful.
16330088732	An action determines the removal of the AuthInfo	K/V, XML	CH, T	An existing AuthInfo was deleted.
33300102900	The provided DNS data are not acceptable. Domain "Status" is "failed".	K/V, XML	C, U, CH	The Nameserver Predelegation Check during registration or update of a domain with status failed, failed.
33300102912	Predelegation Check warning	K/V, XML	C, CH, U	These are warnings issued as a result of the Nameserver Predelegation Check.
46000108827	Secure Entry Point flag should be set in DNSKEY	K/V, XML	C, CH, U	You did not set the corresponding flag for Dnskey.
53000060009	At least one Contact does not exist	K/V, XML	C, CH, U	One of the Contacts stated in the request does not exist.

Error Code	Error Message	Output Format	Request	Brief Description
53000080011	The request cannot be carried out because of an active DISPUTE.	K/V, XML	CH	A holder change is impossible for this domain because it bears an active DISPUTE entry.
53000080012	Domain "Status" is "failed"	K/V, XML	C, T	An error occurred when the Nameserver Predelegation Check was executed in the course of the registration.
53300062003	The value for keyword "Domain" has an invalid format [dynamic error text]	K/V, XML	C, DC	The syntax of the stated domain is invalid.
53300081600	Within a Contact "Email" is missing	K/V, XML	C, U, CH	E-mail is a mandatory field and must always be filled in for PERSON and ORG.
53300100021	Technical Predelegation Check Problem - please try later	K/V, XML	C, U, CH	The Nameserver Predelegation Check failed due to a technical problem.
53300102902	Required glue record is missing [dynamic error text]	K/V, XML	C, U, CH	No IP(s) were stated for the name server(s).

Error Code	Error Message	Output Format	Request	Brief Description
53300102912	Nameserver error [dynamic error text]	K/V, XML	C, U, CH	Name server error: The dynamic part of the error message contains the error message of the Predelegation Check.
53300102920	Occurrences of keyword “Nserver” out of range [min: 2, max: 13]	K/V, XML	C U, CH	“Nserver” was specified more than 13 times or only once.
53300103022	Missing mail exchange preference of DNS entry	K/V, XML	C, U, CH	You did not enter any preference value (0-999) for the MX record.
53380042010	Keyword "Domain" cannot be recognized	K/V, XML	C, U, CH, D, T	The value of the keyword “Domain” was not found.
53380062206	Mismatch between values for "Domain-ace" and "Domain"	K/V, XML	C, U, CH, D, T	Domain and domain-ace do not match.
56300108812	The provided DNS data are not acceptable	K/V, XML	C, U, CH	The DNS data is not accepted.
56310082011	Domain "Status" is "redemptionPeriod”	K/V, XML	C	The domain cannot be registered when in „redemptionPeriod“.

Error Code	Error Message	Output Format	Request	Brief Description
63200020603	This Contact is administered by another provider ([Handle is repeated])	K/V, XML	C, U, CH, D, T	A request is implemented with handles for General Request and Abuse Contact that belong to another RegAcc.
63300022003	This domain is administered by another provider [dynamic error text]	K/V, XML	C, U, CH, D, T, R	The domain is administered by another RegAcc.
63300042100	Keyword "Domain-ace" cannot be recognized	K/V, XML	C, U, CH, D, T	The keyword “Domain-ace” cannot be recognized.
63300042102	Keyword "Domain-ace" can appear once and just once	K/V, XML	C, U, CH, D, T	The keyword “Domain-ace” can be stated only once.
63300043003	Occurrences of keyword "Nsentry" out of range [dynamic error text]	K/V, XML	C, U, CH	Too many entries for "Nsentry"
63300043100	Keyword "Nserver" or "Nsentry" cannot be recognized	K/V, XML	C, U, CH	The keyword "Nserver" or the keyword "Nsentry" was not found.

Error Code	Error Message	Output Format	Request	Brief Description
63300060005	Contact occurs more than once in same role	K/V, XML	C, U, CH	The Contact handle was stated several times for the same role.
63300062009	Domain doesn't exist [repetition of the non-existent domain]	K/V, XML	U, CH, D, T	Domain does not exist.
63300062905	Duplicate value(s) for “Nserver”	K/V, XML	C, U, CH	The values for "Nserver" are identical.
63300063002	Preference values for MX-RR are out of range (0-999)	K/V, XML	C, U, CH	The preference range of MX-RR is 0-999.
63300063003	Incorrect value(s) for “Nsentry”	K/V, XML	C, U, CH	The values for “Nsentry” are invalid.
63300063005	Duplicate value for “Nsentry” record is present	K/V, XML	C, U, CH	The values for "Nsentry" are identical.
63300063006	“Nsentry” has invalid owner	K/V, XML	C, U, CH	The value entered for "Nsentry" is invalid.

Error Code	Error Message	Output Format	Request	Brief Description
63300063011	Nserver has invalid owner [dynamic error text]	K/V, XML	C, U, CH	The value entered for "Nsentry" is invalid.
63300063016	At least 2 MX entries have the identical preference	K/V, XML	C, U, CH	There are MX entries with identical preference.
63300063102	Unknown type of DNS data	K/V, XML	C, U, CH	The type of DNS data is unknown.
63300063103	Illegal type of DNS data	K/V, XML	C, U, CH	The DNS data is not permitted.
63300103005	No other MX resource record is allowed in combination with "Null MX"	K/V, XML	C, U, CH	MX 0 . is not allowed in combination with other MX resource records
63300080013	Too many Contacts are indicated	K/V, XML	C, U, CH	You stated too many Contact handles for one role.

Error Code	Error Message	Output Format	Request	Brief Description
63300080700	Contact with wrong value for "Type" ["HANDLE"]	K/V, XML	C, U, CH	Type and role do not match for one of the Contacts.
63300083005	Value for “Nsentry” contains unknown host	K/V, XML	C, U, CH	The value entered for "Nsentry" is invalid.
63300102913	Incorrect value(s) for “Nserver” [dynamic error text]	K/V, XML	C, U, CH	Invalid name server specified
63310062004	Domain already exists	K/V, XML	C	The domain exists already.

Error Code	Error Message	Output Format	Request	Brief Description
63310062007	Inadmissible value for keyword "Domain" [dynamic error text]	K/V, XML	C, U, CH, R, D, T, DC, DI	**Example** The error code for domain INFO and domain CHECK requests is returned when the domain cannot be converted into an ACE string due to invalid characters or when no second-level domain has been specified. The error message is then: `ERROR: 63310062007 Inadmissible value for keyword "Domain" [Domain does not meet rules for domain name format]` The rules were set according to the IETF standard STD 3.
63450043902	Keyword “Recursive” can appear once and just once	K/V, XML	I	The keyword "Recursive" can be stated only once.
63450063902	Inadmissible value for keyword “Recursive”	K/V, XML	I	The value for "Recursive" is invalid.
66300063003	Missing mail exchange host	K/V, XML	C, U, CH	The mail exchange host (MX) is missing.

Error Code	Error Message	Output Format	Request	Brief Description
66300063006	“Nsentry” has invalid owner	K/V, XML	C, U, CH	The NS-Entry is faulty.
66300048905	Invalid DNS Data (Nsentry, Dnskey)	K/V, XML	C, U, CH	The DNS data is invalid.
66300063023	AAAA resource record requires corresponding A resource record	K/V, XML	C, U, CH	An AAAA-NS-Entry requires a corresponding IN A record, too.
66300068803	The value for keyword "Dnskey" has an invalid format	K/V, XML	C, U, CH	“Dnskey” has invalid format
66300068807	"Dnskey" invalid	K/V, XML	C, U, CH	“Dnskey” is invalid
66310082014	Domain is not in Redemption Grace Period	K/V, XML	R	A RESTORE was sent for a domain that is not in the "redemptionPeriod" status.
83000000004	Validation of XML request against Schema failed	XML	C, U, CH, D, T	Validation of the request against the schema has failed.

Requests Related to Public Contact Details of the Registrar

Overview

Introduction

The registrar contact details is a display of the data of the RegAcc profile, which is limited to those data that represent public information (see RAI).

This chapter describes the data object "public registrar contact details" and the requests you can perform for this data object.

Contents

This chapter covers the following topics:

- The Public Registrar Contact Details at a Glance
- The Public Registrar Contact Details Data Object
- RegAcc INFO Request
- Reply to RegAcc INFO Requests
- Error Messages Related to RegAcc INFO Requests

Public Registrar Contact Details Data Object

Contents of the Data Object

The "Public Registrar Contact Details" data object contains all the relevant, public data of the RegAcc profile:

- RegAccId,
- Name,
- Address,
- Phone number,
- Fax number,
- E-mail address,
- URL,
- MemberAccId,
- General Request and
- Abuse Contact.

Processing Options

With the RegAcc INFO request you can query your own public registrar contact details or those of others.

Public Registrar Contact Details Data Object: Syntax and Processing Rules

The table below informs about the Public Registrar Contact Details:

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
RegAccId	regacc:handle	1	token 7 - 13	regaccid-rule	The RegAc-cId is issued by DENIC and is com-posed of "DENIC-" and a numerical value. It is used to uniquely identify the RegAcc.	-
Name	regacc:name	1	nor-malizedString 1 - 255	Name-rule	This is the holder of the registration account.	Once set, you cannot change the keyword "name" any more.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
Address	regacc:address	1 - *	nor-malizedString 1 - 255	address-rule	Street and house num-ber	-
PostalCode	regac-c:postalCode	0 - 1	token 9 - 32	postalcod-e-rule	This is the postal code.	Some countries do not have postal code system; therefore "0" is a valid value.
City	regacc:city	1	nor-malizedString 1 - 80	city-rule	This is the town or city.	-
CountryCode	regac-c:countryCode	1	enumeration 2	country-erule	This is the applicable country code.	You must use the country code of the ISO-3166-1 alpha-2 List

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
Phone	regacc:phone	1 - *	nor-malizedString 1 - 255	phone-rule	This is the phone num-ber of the registrar. Please use the inter-national format in accordance with the ITU-T E.123 standard (E.123 : Nota-tion for national and international telephone numbers, e-mail addresses and web addresses)	-

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
					The first character must be a "+", and it must be fol-lowed by the country dial-ling code.	

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
Fax	regacc:fax	1 - *	nor-malizedString 1 - 255	phone-rule	This is the fax number of the regis-trar. We recom-mend to use the inter-national format as defined by the ITU-T E.123 stand-ard (E.123 : Notation for national and international telephone numbers, e-mail addresses and web addresses)	-

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
					The first character must be a "+", and it must be followed by the country dialling code.	
Email	regacc:email	1	normalizedString 3 - 255	email-rule see RFC5322 - Internet Message Format	This is the e-mail address of the registrar.	-

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
URL	regacc:url	0 - 1	nor-malizedString 11 - 255	Syntax as defined in RFC 3986 - Uniform Resource Identifier (URI): Gen-eric Syntax	This is the URL of the registrar.	-
Mem-berAccId	regac-c:memberacc	1	token 7 - 13	regaccid-rule	The Mem-berAccId is issued by DENIC and is composed of "DENIC-" and a numerical value. It is used to uniquely identify the MemberAcc.	-

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
Changed	regacc:changed	1	dateTime	date-rule	Date and time format according to ISO 8601 Date and time format	The same format is used for all services to specify the date and time in INFO and WHOIS out-puts. The date and time specified for the RegAcc is updated as soon as the data for General Request and/or Abuse Contact have been changed.
Type	contact:type	0 - 2	enumeration 7	role-erule	Typ of the contact: REQUEST = E-mail address or URL	Once set, the Contact type can-not be changed any more.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
URI-Tem-plate	contact:uri-tem-plate	0 - 2	nor-malizedString 8 - 1024	Syntax as defined in RFC 6570 URI Tem-plate	URI template with the optional vari-ables "Ala-bel" and "Ulabel" that can be used for domains In case of a CREATE request, the content of the URI-Tem-plate is con-verted into a URL or e-mail as a test (For Alabel and Ulabel, a sample domain is used.) to check the syntax on	This is to be used in relation with General Request and Abuse Contact. **Notice** It is man-datory to specify a General Request in the RegAcc profile.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Description	Policy
					correctness.	
Changed	contact:changed	0 - 2	dateTime	date-rule	Date and time format according to ISO 8601 Date and time format	The same format is used for all services to specify the date and time in INFO and WHOIS out-puts.

regaccINFO Request

Function

This request is used to query public registrar contact details of own RegAccs and of those administered by others. You can query your own public registrar contact details as well as those of others.

Requirements

The registrar account must exist.

Request Parameters

A request is composed of the fields and parameters described below:

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
RegAcc	regac-c:handle	1	token 7 - 13	regaccid-rule	The RegAc-cId is issued by DENIC and is com-posed of "DENIC-" and a numerical value. It is used to uniquely identify the RegAcc.
Action	regacc:info	1	enu-meration	info-erule	Type of request
Version	-	1	enu-meration	version-erule	Version, only for requests in key/value format

Examples

- xml
- k/v

format: xml

request: regaccINFO

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0"
xmlns:regacc="http://registry.denic.de/regacc/5.0">
   <regacc:info>
      <regacc:handle>DENIC-99995</regacc:handle>
   </regacc:info>
</registry-request>
```

format: xml

response: regaccINFO

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>eb7c2a09-a963-11e9-aba1-217bf86bcda0</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
      <regacc:infoData xmlns:regacc="h-
ttp://registry.denic.de/regacc/5.0">
        <regacc:handle>DENIC-99995</regacc:handle>
        <regacc:name>DENIC eG - Business Services</regacc:name>
        <regacc:postal>
          <regacc:address>DENIC eG</regacc:address>
          <regacc:address>Kaiserstraße 75-77</regacc:address>
          <regacc:postalCode>60329</regacc:postalCode>
          <regacc:city>Frankfurt</regacc:city>
          <regacc:countryCode>DE</regacc:countryCode>
        </regacc:postal>
        <regacc:phone>+49 69 27235 0</regacc:phone>
```

```
19          <regacc:fax>+49 69 27235 235</regacc:fax>
20          <regacc:email>dbs@denic.de</regacc:email>
21          <regacc:url>http://www.denic.de</regacc:url>
22          <regacc:memberacc>DENIC-99995</regacc:memberacc>
23          <regacc:contact role="generalrequest">
24            <contact:type xmlns:contact="h-
    ttp://registry.denic.de/contact/5.0">REQUEST</contact:type>
25            <contact:uri-template xmlns:contact="h-
    ttp://registry.denic.de/contact/5.0">mailto:-
    general@denic.de?subject=domain:{Ulabel}</contact:uri-template>
26            <contact:changed xmlns:contact="h-
    ttp://registry.denic.de/contact/5.0">2019-01-01T01:00:00+01:00</con-
    tact:changed>
27          </regacc:contact>
28          <regacc:contact role="abusecontact">
29            <contact:type xmlns:contact="h-
    ttp://registry.denic.de/contact/5.0">REQUEST</contact:type>
30            <contact:uri-template xmlns:contact="h-
    ttp://registry.denic.de/contact/5.0">mailto:-
    abuse@denic.de?subject=domain:{Ulabel}</contact:uri-template>
31            <contact:changed xmlns:contact="h-
    ttp://registry.denic.de/contact/5.0">2019-01-01T01:00:00+01:00</con-
    tact:changed>
32          </regacc:contact>
33          <regacc:changed>2019-05-17T10:48:29+02:00</regacc:changed>
34        </regacc:infoData>
35      </tr:data>
36    </tr:transaction>
37  </registry-response>
```

format: k/v

request: regaccINFO

```
1  Version: 5.0
2  Action: info
3  Regacc: DENIC-99995
```

format: k/v

response: regaccINFO

```
Result: success
STID: c69ae650-a962-11e9-aba1-217bf86bcda0
Regaccid: DENIC-99995
Name: DENIC eG - Business Services
Address: DENIC eG
Address: Theodor-Stern-Kai 1
Postalcode: 60596
City: Frankfurt am Main
Countrycode: DE
Phone: +49 69 27235 0
Fax: +49 69 27235 238
Email: dbs@denic.de
Url: http://www.denic.de
Memberaccid: DENIC-99995
Changed: 2019-05-17T10:48:29+02:00

[General Request]
Type: request
Uri-template: mailto:general@denic.de?subject=domain:{Ulabel}
Changed: 2019-01-01T01:00:00+01:00

[Abuse Contact]
Type: request
Uri-template: mailto:abuse@denic.de?subject=domain:{Ulabel}
Changed: 2019-01-01T01:00:00+01:00
```

Reply to regaccINFO Requests

Features

The URI templates for General Request and Abuse Contact are not expanded.

General Request and Abuse Contact are output only if corresponding values are stored in the RegAcc profile (RAI).

Description of Data Fields

The reply of the RRI to a RegAcc INFO request comprises the data fields described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
RegAccId	regacc:handle	1	token 7 - 13	regaccid-rule	The RegAccId is issued by DENIC and is composed of "DENIC-" and a numerical value. It is used to uniquely identify the RegAcc.
Name	regacc:name	1	normalizedString 1 - 255	Name-rule	This is the holder of the registration account.
Address	regacc:address	1 - *	normalizedString 1 - 255	address-rule	Street and house number

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
PostalCode	regacc:postalCode	0 - 1	token 9 - 32	postalcode-rule	This is the postal code.
City	regacc:city	1	normalizedString 1 - 80	city-rule	This is the town or city.
CountryCode	regacc:countryCode	1	enumeration 2	country-erule	This is the applicable country code.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Phone	regacc:phone	1 - *	normalizedString 1 - 255	phone-rule	This is the phone number of the registrar. Please use the international format in accordance with the ITU-T E.123 standard (E.123 : Notation for national and international telephone numbers, e-mail addresses and web addresses) The first character must be a "+", and it must be followed by the country dialling code.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Fax	regacc:fax	1 - *	normalizedString 1 - 255	phone-rule	This is the fax number of the registrar. We recommend to use the international format as defined by the ITU-T E.12323 standard (E.123 : Notation for national and international telephone numbers, e-mail addresses and web addresses) The first character must be a "+", and it must be followed by the country dialling code.
Email	regacc:email	1	normalizedString 3 - 255	email-rule see RFC5322 - Internet Message Format	This is the e-mail address of the registrar.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
URL	regacc:url	0 - 1	normalizedString 11 - 255	Syntax as defined in RFC 3986 (Uniform Resource Identifier (URI): Generic Syntax)	This is the URL of the registrar.
MemberAccId	regacc:memberacc	1	token 7 - 13	regaccid-rule	The MemberAccId is issued by DENIC and is composed of "DENIC-" and a numerical value. It is used to uniquely identify the MemberAcc.
Changed	regacc:changed	1	dateTime	date-rule	Date and time format according to ISO standard ISO 8601 Date and time format
Type	contact:type	0 - 2	enumeration 7	role-erule	The data of the field is output only when General Request and/or Abuse Contact are included in the RegAcc profile.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
URI-Template	contact:uri-template	1	normalizedString 8 - 1024	Syntax as defined in RFC 6570 URI Template	The data of the field is output only when General Request and/or Abuse Contact are included in the RegAcc profile.
Changed	contact:changed	-	dateTime	date-rule	The data of the field is output only when General Request and/or Abuse Contact are included in the RegAcc profile.

Error Messages Related to RegAcc Requests

Potential Error Messages

Below you find a list of the error messages that will occur exclusively during the processing of a RegAcc request.

Comments on the Table

The column "Output Format" informs whether the error message may occur with the key/value (= K/V) and / or with the XML format (= XML).

Error Code	Error Message	Output Format	Brief Description
63310062008	Inadmissible value for keyword "RegAcc" [Repeat of the user entry]	K/V, XML	Specification of an incorrect value
63300062010	RegAcc doesn't exist [Wiederholung RegAcc]	K/V, XML	The RegAcc does not exist.

Provider Change Requests

Overview

Introduction

A .de domain is administered through a registration account (RegAcc) of a DENIC member. If the domain holder wants to transfer administration from the current RegAcc to another RegAcc, they may ask their current provider to issue an AuthInfo1. Alternatively, the new (future) provider can request DENIC to issue an AuthInfo2.

You can then use this AuthInfo1or AuthInfo2 to request a provider change.

With an AuthInfo2, any other RegAcc can restore a domain that is in the Redemption Grace Period status.In this chapter, we will describe the request types you need to prepare and execute a provider change.

Contents

This section covers the following topics:

- Chprov Data Object,
- CREATE-AUTHINFO1 Request,
- DELETE-AUTHINFO1 Request,
- CREATE-AUTHINFO2 Request,
- CHPROV Request,
- Replies to CREATE-AUTHINFO1, CREATE-AUTHINFO2, CHPROV Requests and
- Error Messages Related to Provider Change Requests.

Chprov Data Object

Contents of the Data Object

The "Chprov" data object is derived from the "Domain" data object and the information provided for the AuthInfo.

Processing Options

You may record AuthInfos in DENIC's database, delete them or execute a provider change.

Chprov Data Object: Syntax and Processing Rules

The table below informs you in which way the "Chprov" data object supplements the "Domain" data object:

K/V Key-word	XML Namesp-ace and Element	Occur-rence min - max	Type / Lengt-h	Value Range	Descrip-tion	Policy
AuthInfo	authinfo	1	token 1 - 255	authinfo-rule	An AuthInfo-1 or AuthInfo-2 is stated.	The trans-ferred AuthInfo must match an AuthInfo-1 or AuthInfo-2 stored in the DENIC data-base.
AuthIn-foHash	hash	1	token 64	auth-infohash-rule	This is the hash value of the stored AuthInfo-1.	-
AuthIn-foExpire	expire	0 - 1	token 8	digit-rule	This is the term of validity of the AuthInfo-1.	The EXPIRE date can be 30 days ahead at max-imum. The default is the cur-rent date + 30 cal-endar days.

domainCREATE-AUTHINFO1 Request

Function

This request is used to store an AuthInfo for a domain.

Maximum validity of an AuthInfo is 30 days. Upon expiration of this period (AuthInfoExpire) the AuthInfo will be deleted automatically.

Requirements

- Only one AuthInfo1 can be stored for each domain.
- The AuthInfo1 has to be between 8 and 16 characters long.
- You can store an AuthInfo1 in addition to an existing AuthInfo2.
- Only the administering RegAcc is entitled to store an AuthInfo1.
- If an AuthInfo is stored for a "failed" domain, the EXPIRE date of the domain is communicated in the AuthInfo notification by e-mail.
- You must create the hash by means of the SHA-256 algorithm. No Salt and just one iteration are allowed.
- The output of the hash function (256 bit) must be encrypted hexadecimally in the request.

Request Parameters

A CREATE-AUTHINFO1 request is composed of the "Chprov" data object fields and some other parameters, which are described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	domain:createAuthInfo1	1	enumeration	creauthinfo1-erule	This is the request type.

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique trans-action ID which is issued by the client.

Common Mistakes

- The stated domain is not administered by the RegAcc submitting the request.
- Encryption of the AuthInfo hash is invalid (not SHA256).
- An AuthInfo1 already exists for the domain.

Features

An existing AuthInfo2 will persist.

Examples

- xml
- k/v

format: xml

format: xml**request:** domainCREATE-AUTHINFO1

request: domainCREATE-AUTHINFO1

format: xml

response: domainCREATE-AUTHINFO1

status: connect

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "connect"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the
holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
    <tr:message level="info" code="16350000040">
      <tr:text>Verification information must be provided for the
holder(s) to avoid dedelegation by</tr:text>
      <tr:argument>2024-12-09T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: xml

response: domainCREATE-AUTHINFO1

status: serverHold

```
<?xml version="1.0" encoding="UTF-8"?>
```

format: xml**response:** domainCREATE-AUTHINFO1**status:** serverHold

```
 2
 3  <registry-response xmlns="http://registry.denic.de/global/5.0"
    xmlns:tr="http://registry.denic.de/transaction/5.0">
 4    <tr:transaction>
 5      <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
 6      <tr:ctid>cba-987654321</tr:ctid>
 7      <tr:result>success</tr:result>
 8      <tr:message level="info" code="53000080013">
 9        <tr:text>Domain "Status" is "serverHold"</tr:text>
10      </tr:message>
11      <tr:message level="info" code="16350000041">
12        <tr:text>Verification information must be provided for the
    holder(s) to avoid deletion by</tr:text>
13        <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
14      </tr:message>
15    </tr:transaction>
16  </registry-response
```

format: k/v

request: domainCREATE-AUTHINFO1

```
1  Version: 5.0
2  Action: create-authinfo1
3  Ctid: cba-987654321
4  Domain: de-example.de
5  Authinfohash: 4213d924230224f-
   d719218b4acbd92f96ebe4344f3d5d1478dede1aa44e4cf4b
6  Authinfoexpire: 20100724
```

format: k/v

response: domainCREATE-AUTHINFO1

status: connect

```
1  Result: success
2  STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d
3
4  INFO: 53000080013 Domain "Status" is "connect"
```

```
5 | INFO: 16350000040 Verification information must be provided for the
  | holder(s) to avoid dedelegation by [2024-12-09T15:45:01+01:00]
6 | INFO: 16350000041 Verification information must be provided for the
  | holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
7 | Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
8 | Ctid: cba-9345345321
```

format: k/v

response: domainCREATE-AUTHINFO1

status: serverHold

```
1 | Result: success
2 | STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d
3 |
4 | INFO: 53000080015 Domain "Status" is "serverHold"
5 | INFO: 16350000041 Verification information must be provided for the
  | holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
6 | Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
7 | Ctid: cba-9345345321
```

domainCREATE-AUTHINFO2 Request

Function

The request is used to create an AuthInfo2 for a domain.

Any DENIC member can request a generation of an AuthInfo2 for a domain. DENIC then generates a password and sends it by post to the current domain owner at the address stored in DENIC's database.

Requirements

- The AuthInfo2 has to be between 8 and 16 characters long.
- The hash must be created using only the SHA-256 algorithm, without a salt and with a single iteration.

Request Parameters

A request consists of the fields of the “Chprov” data object and the following additional parameters, which are described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	domain:createAuthInfo2	1	enumeration	creauthinfo2-erule	This is the request type.
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

An AuthInfo2 has already been stored.

Features

- An existing AuthInfo1 will persist.
- The administering RegAcc receives a message (either via RRI queue or via e-mail) informing it that an AuthInfo2 has been created for one of its domains.
- Only one AuthInfo2 can be stored for each domain.
- All domain holders receive a letter. The AuthInfo2 letters are sent worldwide.

- All domain holders receive the same Authinfo2.
- Maximum validity of an AuthInfo2 is 30 days. Upon expiration of this period, the AuthInfo will be deleted automatically.
- When the domain is deleted, an existing AuthInfo2 for the domain is also deleted.
- An AuthInfo2 can be stored for any existing domain independent of its status ("connect","failed" or “redemptionPeriod”).
- If an AuthInfo2 is deposited for a domain in the “failed” status, the corresponding notification letter will indicate the expiration date of the domain.
- If an AuthInfo2 is stored for a domain with „redemptionPeriod“ status, the term of the Redemption Grace Period is set anew to 30 days.
- For a domain in the status "redemptionPeriod", an AuthInfo2 can only be requested until midnight on the day of the purge release.

> **Notice**
>
> Example:
>
> Redemption Grace Period = 30 Days
>
> 1. August: DELETE
>
> 20. August: CREATE-AUTHINFO2
>
> ~~31. August: Ende der Redemption Grace Period~~
>
> 19. September: Ende der Redemption Grace Period
>
> Text

Examples

- xml
- k/v

format: xml

format: xml**request:** CREATE-AUTHINFO2

request: CREATE-AUTHINFO2

format: xml

response: domainCREATE-AUTHINFO2

status: connect

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "connect"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the
holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
    <tr:message level="info" code="16350000040">
      <tr:text>Verification information must be provided for the
holder(s) to avoid dedelegation by</tr:text>
      <tr:argument>2024-12-09T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: k/v

response: domainCREATE-AUTHINFO2

status: serverHold

```
<?xml version="1.0" encoding="UTF-8"?>
```

format: k/v**response:** domainCREATE-AUTHINFO2**status:** serverHold

```

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "serverHold"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the
holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: k/v

request: CREATE-AUTHINFO2

```
Version: 5.0
Action: create-authinfo2
CTID: cba-987654321
Domain: de-example.de
```

format: k/v

response: domainCREATE-AUTHINFO2

status: connect

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

INFO: 53000080013 Domain "Status" is "connect"
INFO: 16350000040 Verification information must be provided for the
holder(s) to avoid dedelegation by [2024-12-09T15:45:01+01:00]
INFO: 16350000041 Verification information must be provided for the
holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
```

```
Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
Ctid: cba-9345345321
```

format: k/v

response: domainCREATE-AUTHINFO2

status: serverHold

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

INFO: 53000080015 Domain "Status" is "serverHold"
INFO: 16350000041 Verification information must be provided for the
holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
Ctid: cba-9345345321
```

domainDELETE-AUTHINFO1 Request

Function

This request is used to delete a stored AuthInfo1 during its term of validity.

Requirements

- An AuthInfo1 must be stored for the domain.
- Only the RegAcc administering the domain is entitled to make the request.

Request Parameters

A request consists of the fields of the “Chprov” data object and the following additional parameters, which are described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	domain:deleteAuthInfo1	1	enumeration	delauthinfo1-erule	This is the request type.
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

- The stated domain is not administered by the RegAcc submitting the request.
- No AuthInfo or an AuthInfo2 is stored for the domain.

Features

An existing AuthInfo2 will persist.

Examples

- xml
- k/v

format: xml

format: xml**request:** CREATE-AUTHINFO2

request: CREATE-AUTHINFO2

format: xml

response: domainCREATE-AUTHINFO2

status: connect

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "connect"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
    <tr:message level="info" code="16350000040">
      <tr:text>Verification information must be provided for the holder(s) to avoid dedelegation by</tr:text>
      <tr:argument>2024-12-09T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: k/v

response: domainCREATE-AUTHINFO2

status: serverHold

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

format: k/v**response:** domainCREATE-AUTHINFO2**status:** serverHold

```
 2
 3  <registry-response xmlns="http://registry.denic.de/global/5.0"
    xmlns:tr="http://registry.denic.de/transaction/5.0">
 4    <tr:transaction>
 5      <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
 6      <tr:ctid>cba-987654321</tr:ctid>
 7      <tr:result>success</tr:result>
 8      <tr:message level="info" code="53000080013">
 9        <tr:text>Domain "Status" is "serverHold"</tr:text>
10      </tr:message>
11      <tr:message level="info" code="16350000041">
12        <tr:text>Verification information must be provided for the
    holder(s) to avoid deletion by</tr:text>
13        <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
14      </tr:message>
15    </tr:transaction>
16  </registry-response
```

format: k/v

request: CREATE-AUTHINFO2

```
1  Version: 5.0
2  Action: create-authinfo2
3  CTID: cba-987654321
4  Domain: de-example.de
```

format: k/v

response: domainCREATE-AUTHINFO2

status: connect

```
1  Result: success
2  STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d
3
4  INFO: 53000080013 Domain "Status" is "connect"
5  INFO: 16350000040 Verification information must be provided for the
   holder(s) to avoid dedelegation by [2024-12-09T15:45:01+01:00]
6  INFO: 16350000041 Verification information must be provided for the
   holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
```

```
Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
Ctid: cba-9345345321
```

format: k/v

response: domainCREATE-AUTHINFO2

status: serverHold

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

INFO: 53000080015 Domain "Status" is "serverHold"
INFO: 16350000041 Verification information must be provided for the
holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
Ctid: cba-9345345321
```

domainCHPROV Request

Function

The CHPROV request is used to transfer a domain to one's own RegAcc for administration.

Requirements

- You can request a CHPROV only for domains which are administered by another RegAcc or the domains has been deletedand has the the „redemptionPeriod“ status. In this case, the domain is restored through a CHPROV with AuthInfo2 for the RegAcc that has submitted the request.
- Note that you can use only Contact handles of the requesting RegAcc in CHPROV requests. The handles must exist and fulfil the applicable conditions.
- The AuthInfo transferred with the CHPROV request must be identical with an AuthInfo1 or AuthInfo2 stored in DENIC's database.

Request Parameters

A CHPROV request is composed of the "Chprov" data object fields and of some other parameters, which are described below:

K/V Key-word	XML Namespace and Element	Occur-rence min - max	Type / Length	Value Range	Descrip-tion
Action	domain:ch-prov	1	enu-meration	chprov-erule	This is the request type.
Version	-	1	enu-meration	version-erule	This is the version. The keyword is only rel-evant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any vis-ible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.

Common Mistakes

- The AuthInfo stated in the request does not match the AuthInfo(s) stored with DENIC.
- The domain is already under its own administration.
- The provider change shall be combined with a holder change, but the domain is bearing a DISPUTE entry.

Features

- A CHPROV may be combined with a holder change.
- If a CHPROV is executed by using an AuthInfo1, the existing AuthInfo1 will be deleted upon execution of the request; a possibly existing AuthInfo2 will be maintained if the Domainholder remains unchanged.
- If a CHPROV is executed by using an AuthInfo2, the Authinfo2 as well as any possibly existing AuthInfo1 will be deleted.
- If a DISPUTE has been placed on the domain concerned, a CHPROV request cannot be combined with a holder change.
- Domains can be restored from the Redemption Grace Period by means of a CHPROV with AuthInfo2.

Examples

- xml
- k/v

format: xml

request: domainCHPROV

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:domain="http://registry.denic.de/domain/5.0" xmlns:dnsentry="http://registry.denic.de/dnsentry/5.0">
  <domain:chprov>
    <domain:handle>de-example.de</domain:handle>
    <domain:contact role="holder">DENIC-1000002-MAX</domain:contact>
    <domain:contact role="abusecontact">DENIC-1000002-ABUSE</domain:contact>
    <domain:contact role="generalrequest">DENIC-1000002-GENERAL</domain:contact>
    <dnsentry:dnsentry xsi:type="dnsentry:NS">
    <dnsentry:owner>de-example.de</dnsentry:owner>
    <dnsentry:rdata>
      <dnsentry:nameserver>ns1.provider.de</dnsentry:nameserver>
```

format: xml**request:** domainCHPROV

```
    </dnsentry:rdata>
    </dnsentry:dnsentry>
    <dnsentry:dnsentry xsi:type="dnsentry:NS">
    <dnsentry:owner>de-example.de</dnsentry:owner>
    <dnsentry:rdata>
      <dnsentry:nameserver>ns2.provider.de</dnsentry:nameserver>
    </dnsentry:rdata>
    </dnsentry:dnsentry>
    <domain:authInfo>secret</domain:authInfo>
    <dnsentry:dnsentry xsi:type="dnsentry:DNSKEY">
      <dnsentry:owner>de-example.de.</dnsentry:owner>
        <dnsentry:rdata>
          <dnsentry:flags>257</dnsentry:flags>
          <dnsentry:protocol>3</dnsentry:protocol>
          <dnsentry:algorithm>8</dnsentry:algorithm>
          <dnsentry:publicKey>AwEAAa-
jpN1r-
jU365IDVSd+-
duoMg4UF4S8zBSpJ4D/p2VSOKYVN7/0CkAV6pOh-
mdcoUMETA2LDPBWjEh4r-
rGiBN9ReRw/4udM-
giEf3YyvsJEDpF6vtrWzqe+Cu9i-
u8QV//8ESlQgqK8TqVcR7e/FPd89R44S68GhZRFGw7h7sDrRnGIZf
</dnsentry:publicKey>
        </dnsentry:rdata>
    </dnsentry:dnsentry>
  </domain:chprov>
  <ctid>cba-987654321</ctid>
</registry-request>
```

format: xml

response: domainCHPROV

status: connect

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0"
xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
```

format: xml**response:** domainCHPROV**status:** connect

```
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "connect"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
    <tr:message level="info" code="16350000040">
      <tr:text>Verification information must be provided for the holder(s) to avoid dedelegation by</tr:text>
      <tr:argument>2024-12-09T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: xml

response: domainCHPROV

status: serverHold

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>538f1687-963a-11df-a9e2-9519b5688882</tr:stid>
    <tr:ctid>cba-987654321</tr:ctid>
    <tr:result>success</tr:result>
    <tr:message level="info" code="53000080013">
      <tr:text>Domain "Status" is "serverHold"</tr:text>
    </tr:message>
    <tr:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the holder(s) to avoid deletion by</tr:text>
      <tr:argument>2024-12-16T15:45:01+01:00</tr:argument>
    </tr:message>
  </tr:transaction>
</registry-response
```

format: k/v

request: domainCHPROV

```
Version: 5.0
Action: chprov
CTID: cba-987654321
Domain: de-example.de
Holder: DENIC-1000002-HOLDER
Generalrequest: DENIC-1000002-GENERAL
Abusecontact: DENIC-1000002-ABUSE
Nserver: ns1.beispiel-eins.de
Nserver: ns2.beispiel-eins.de
Dnskey: 257 3 8 AwEAA-
coFUSy-
g1mkE5c33q8UbDiRZx5+/QtqFjVcyTEd-
v7YBYp9Un-
qrbXr7g4p8aDMI0ZuN4M8bxlPz+ItVfWO71rSk-
cxK1HwqmH4Pi1vSM3L6uYqZopEG9gJLqNpBBmzR29iSwR86Td-
nGUJ21Jfagc/+9xk3xmtdzNK3ROUcn/f8yiBN
Authinfo: secret
```

format: k/v

response: domainCHPROV

status: connect

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

INFO: 53000080013 Domain "Status" is "connect"
INFO: 16350000040 Verification information must be provided for the
holder(s) to avoid dedelegation by [2024-12-09T15:45:01+01:00]
INFO: 16350000041 Verification information must be provided for the
holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
Ctid: cba-9345345321
```

format: k/v

response: domainCHPROV

status: serverHold

```
Result: success
STID: 74ba5119-74d1-4ba6-935f-4feb07ddd14d

INFO: 53000080015 Domain "Status" is "serverHold"
INFO: 16350000041 Verification information must be provided for the
holder(s) to avoid deletion by [2024-12-16T15:45:01+01:00]
Stid: 57ebbb10-9590-11df-a9e2-9519b5688882
Ctid: cba-9345345321
```

Replies to CREATE-AUTHINFO1, CREATE-AUTHINFO2, CHPROV

Description of Data Fields

The replies of the .de registration system to CREATE-AUTHINFO1, CREATE-AUTHINFO2 or CHPROV requests comprise the data fields described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
StId	tr:stid	1	token 30-255	stid-rule	This is a Server Transaction ID (StId) which is assigned by the server. The StId ensures that the request from the provider and DENIC's reply are referenced with a globally unique ID.
CtId	tr:ctid	0 - 1	token 3 - 64	Any visible Unicode char-acter (accord-ing to Unicode ver-sion 3.1)	This is the unique transaction ID which is issued by the client.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	tr:result	1	token	result-erule	**Result of processing:** success = request was executed failed = request was not executed exception = system failure, request was not executed
Info	tr:message level-l="info"	0 - *	normalizedString 1 - 255	message-rule	Information about the request, consisting of a message code and text
Warning	tr:message level-l="warning"	0 - *	normalizedString 1 - 255	message-rule	Warning about the request, consisting of a message code and text
Error	tr:message level-l="error"	0 - *	normalizedString 1 - 255	message-rule	Error message about the request, consisting of a message code and text
AuthInfoExpire	expire	0-1	token 25	date-rule	Expire date in the format of ISO 8601 Date and time format.

Error Messages Related to Provider Change Requests

Potential Error Messages

Below you will find a list of the error messages that may occur in relation with the processing of a provider change requests.

For general error messages which may occur in relation with provider change requests refer to the Annex.

Comments on the Table

- The column "Input Format" informs whether the error message may occur with the key/value (= KV) and / or with the XML format (= XML).
    - The column "Request type" informs about the types of requests in relation with which the error message may occur:
        - CHPROV (=C)
        - CREATE-AUTHINFO1 (=CA1)
        - CREATE-AUTHINFO2 (=CA2)
        - DELETE-AUTHINFO1 (=DA1)

Error Code	Error Message	Input Format	Request Type	Brief Description
33300102900	The provided DNS data are not acceptable. Domain "Status" is "failed".	KV / XML	C	The Nameserver Predelegation Check of a domain with status failed, failed.
33300102912	Predelegation Check warning	KV / XML	C	These are warnings issued as a result of the Nameserver Predelegation Check.
46000108827	Secure Entry Point flag should be set in DNSKEY	KV / XML	C	You did not set the corresponding flag for Dns-key.
53000060009	At least one Contact does not exist	KV / XML	C	One of the Contacts stated in the request does not exist.
53300081600	Within a Contact "Email" is missing	KV / XML	C	The Contact contains no e-mail address.
53300100021	Technical Predelegation Check Problem - please try later	KV / XML	C	The Nameserver Predelegation Check failed due to a technical problem. Please submit an Update for the domain.
53300102902	Required glue record is missing [dynamic error text]	KV / XML	C	No IP(s) was/were stated for the name server(s).

Error Code	Error Message	Input Format	Request Type	Brief Description
53300102912	Nameserver error [dynamic error text]	KV / XML	C	Name server error: The dynamic part of the error message contains the error message of the Predelegation Check.
53300102912	Nameserver error [dynamic error text]	KV / XML	C	The Nameserver Predelegation Check failed.
53300102920	Occurrences of keyword “Nserver” out of range [min: 2, max: 13]	KV / XML	C	“Nserver” was stated too often or once.
53300103022	Missing mail exchange preference of DNS entry	KV / XML	C	You did not enter any preference value (0-999) for the MX record.
53370080005	Required change provider request data is missing	KV	C	You did not include all mandatory data.
53380042010	Keyword "Domain" cannot be recognized	KV / XML	C	The value of the keyword “Domain” was not found.
53380062206	Mismatch between values for "Domain-ace" and "Domain"	KV / XML	C	Domain and domain-ace do not match.

Error Code	Error Message	Input Format	Request Type	Brief Description
53380062206	Mismatch between values for "Domain-ace" and "Domain"	KV / XML	C / U / CH / D / T	Domain and domain-ace do not match.
54300062506	Holder does not match	KV / XML	C	Domain holder does not match.
56000088709	Specified AuthInfo2 does not exist	KV / XML	C	The stated AuthInfo2 does not exist.
56300108812	The provided DNS data are not acceptable	KV / XML	C	The DNS data is not accepted.
63200020603	This Contact is administered by another provider ([repetition of the handle])	KV / XML	C / U / CH / D / T	A request is implemented with handles for General Request and Abuse Contact that belong to another RegAcc.
63300022003	This domain is administered by another provider [dynamic error text]	KV / XML	C	The domain is administered by another RegAcc.
63300042100	Keyword "Domain-ace" cannot be recognized	KV / XML	C	The keyword "Domain-ace" cannot be found.
63300042102	Keyword "Domain-ace" can appear once and just once	KV / XML	C	The keyword "Domain-ace" can be stated only once.

Error Code	Error Message	Input Format	Request Type	Brief Description
63300043003	Occurrences of keyword "Nsentry" out of range [dynamic error text]	KV / XML	C	Too many entries for "Nsentry". The maximum number of permitted NS entries is five.
63300043100	Keyword "Nserver" or "Nsentry" cannot be recognized	KV / XML	C	The keyword "Nserver" or the keyword "Nsentry" was not found.
63300060005	Contact occurs more than once in same role	KV / XML	C	The Contact handle was stated several times for the same role.
63300062009	Domain doesn't exist [repetition of the non-existent domain]	KV / XML	C / CA1 / CA2 / DA1	Domain does not exist.
63300062905	Duplicate value(s) for “Nserver”	KV / XML	C	The values for "Nserver" are identical.
63300063002	Preference values for MX-RR are out of range (0-999)	KV / XML	C	The preference range of MX-RR is 0-999.
63300063005	Incorrect value(s) for “Nsentry”	KV / XML	C	The value for “Nsentry” is invalid.
63300063005	Duplicate value for “Nsentry” record is present	KV / XML	C	The values for "Nsentry" are identical.
63300063006	“Nsentry” has invalid owner	KV / XML	C	The value entered for "Nsentry" is invalid.

Error Code	Error Message	Input Format	Request Type	Brief Description
63300063011	Nserver has invalid owner [dynamic error text]	KV / XML	C	The value entered for "Nsentry" is invalid.
63300063016	At least 2 MX entries have the identical preference	KV / XML	C	There are MX entries with identical preference values.
63300063102	Unknown type of DNS data	KV / XML	C	The type of DNS data is unknown.
63300063103	Illegal type of DNS data	KV / XML	C	The DNS data is not permitted.
63300080013	Too many Contacts are indicated [dynamic error text]	KV / XML	C	You stated too many Contact handles for one role.
63300080700	Contact with wrong value for "Type" [“HANDLE”]	KV / XML	C	Type and role do not match for one of the Contacts.
63300083005	Value for “Nsentry” contains unknown host	KV / XML	C	The value entered for "Nsentry" is invalid.
63300102913	Incorrect value(s) for “Nserver” [dynamic error text]	KV / XML	C	You stated an invalid Nameserver.

Error Code	Error Message	Input Format	Request Type	Brief Description
63300103005	No other MX resource record is allowed in combination with "Null MX"	KV / XML	C / U / CH	MX 0 . is not allowed in combination with other MX resource records.
63310062007	Inadmissible value for keyword "Domain" [dynamic error text]	KV / XML	C	The value entered for "Domain" is invalid.
63370048700	Keyword "AuthInfo" cannot be recognized [dynamic error text]	KV / XML	C	The keyword "AuthInfo" was not found.
63370088701	AuthInfo does not match	KV / XML	C / I	AuthInfo does not match.
63380063209	Illegal operation on own domain	KV / XML	C	You tried to start a CHPROV for your own domain.
63391048502	Keyword "AuthInfoHash" must appear once and just once	KV / XML	CA1	The keyword "AuthInfoHash" must not be stated more than once.
63391048602	Keyword "AuthInfoExpire" can appear once and just once	KV / XML	CA1	The keyword "AuthInfoExpire" must not be stated more than once.
63391068503	The value for keyword "AuthInfoHash" has an invalid format	KV / XML	CA1	The format used for the "AuthInfoHash" is invalid.

Error Code	Error Message	Input Format	Request Type	Brief Description
63391068603	The value for keyword "AuthInfoExpire" has an invalid format	KV / XML	CA1	The format used for "AuthInfoExpire" is invalid.
63391068704	AuthInfo1 already exists	KV / XML	CA1	AuthInfo1 already exists.
63392068709	AuthInfo1 does not exist	KV / XML	DA1	AuthInfo1 does not exist.
63393068704	AuthInfo2 already exists	KV / XML	CA2	AuthInfo2 already exists.
66300048905	Invalid DNS Data (Nsentry, Dnskey)	KV / XML	C	The DNS data is invalid.
66300068803	The value for keyword "Dnskey" has an invalid format	KV / XML	C	“Dnskey” has invalid format.
66300068807	"Dnskey" invalid	KV / XML	C	“Dnskey” is invalid.

Requests Related to the Message Queue

Introduction

RRI provides information in a message queue for RegAccs. The messages can be the direct result of a request (e.g. message type “ChprovAuthInfo” for a CHPROV request), be provided for a request after a delay (e.g. the “Expire” message type for a newly registered domain that was deleted after 30 days due to incorrect nameserver information) or DENIC creates a message based on a process (e.g. the “DomainDelete” message type, when a domain is deleted by DENIC's legal department).

Contents

This section contains the following topics:

- Request: QUEUE-READ
- Request: QUEUE-DELETE
- Response: QUEUE-READ and QUEUE-DELETE
- Error messages for QUEUE-READ and QUEUE-DELETE requests

Overview of Message Types

Message Type	Description
authInfo2Delete	Message about the deletion of an AuthInfo2 by DENIC because the letter with the AuthInfo2 password could not be delivered
authInfo2Notify	Message about the generation of an AuthInfo2.
authInfoExpire	Message about the expiry date of an AuthInfo1 and AuthInfo2 password.
chprovAuthInfo	In the case of a CHPROV request, the RegAcc from which a domain is moved receives a message about the successful move.
domainDelete	Message about a domain deletion carried out by DENIC's legal department.

Message Type	Description
expire	Message about the date on which the domain expired.
expireWarning	Message about the date on which a domain will expire. The message is sent one week before the expiration date.
contactDelete	Message listing the contact handle IDs for which the contact objects have been deleted due to expired retention periods.

Configuring Message Reception

The receipt of messages can be set in the RAI for a RegAcc profile. In the “MsgFeed” line, you can set:

- an be set in the RAI for a RegAcc profile. In the “MsgFeed” line, you can set:
- “rri”, in which case the messages must be read with QUEUE-READ and deleted from the message queue with QUEUE-DELETE, or
- “mail” to receive the messages automatically by e-mail. The e-mail address from the “Email” line is used for receipt.

> **Notice**
>
> For technical reasons, the messages can also be called up briefly via QUEUE-READ in the message queue when the “mail” setting is selected. DENIC empties the message queue at regular intervals and sends the messages by e-mail.
>
> When using the “mail” setting, you should refrain from reading messages from the message queue in addition to receiving them automatically by e-mail. The response formats differ in their structure and mixing them could cause unwanted processing errors on the part of the RegAccs.

Automatic Email Reminder for older Messages in the Message Queue

- The RegAcc receives an automatic email reminder that there are messages in the message queue that are older than three days.
- The reminder is sent to the e-mail address stored in the “EMailTo” line from the RegAcc profile.
- The reminder is sent once a day if messages older than three days remain in the message queue.

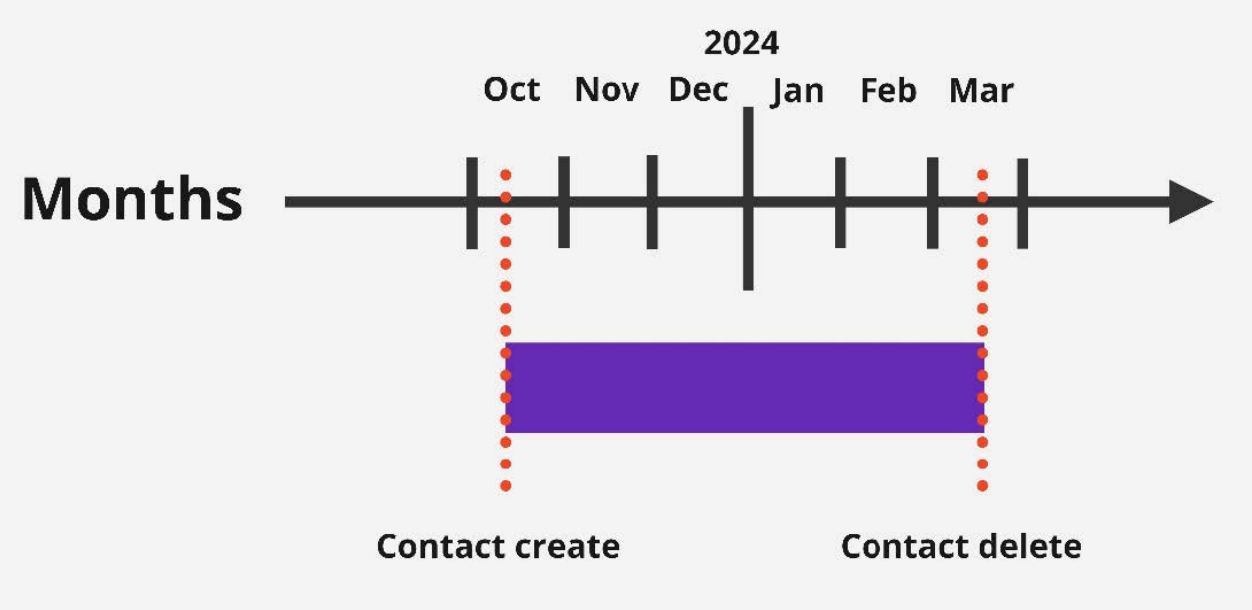
Automatic deletion of Contact Data

- Contact data is automatically deleted after the retention periods have expired.
- The message queue and e-mail are used to inform me of the deletions using the message type “contactDelete”.
- Several contact deletions can be combined in a “contactDelete” message. A maximum of 100 contact handles can be specified in a “contactDelete” message for contacts that have been deleted.

Storage Period for Contact Data

6 Months

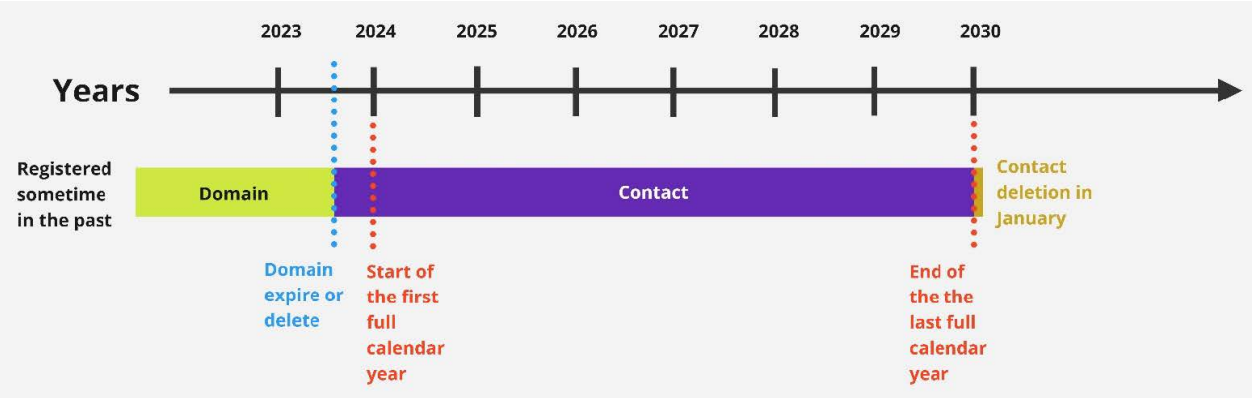
for contacts that have never been referenced or used with a domain



Example for 6 Months	
October 2023	DENIC-1000002-HOLDER is created
October 2023 to March 2024	DENIC-1000002-HOLDER is never used with a domain
March 2024	DENIC-1000002-HOLDER is deleted

6 full Calendar Years

for contacts that were previously referenced with a domain



Example for 6 full Calendar Years	
October 2023	All domains for DENIC-1000002-HOLDER have been deleted or have expired.
have expired. Beginning of 2024 until the end of 2030	Beginning of 2024 until the end of 2030 DENIC-1000002-HOLDER continues to exist for 6 full calendar years.
January 2030	DENIC-1000002-HOLDER no longer has a reference to a domain and is deleted in January.

QUEUE-READ Request

Function

QUEUE-READ requests are used to collect or read messages from the message queue of the RRI from the RegAcc.

Requirements

Messages can only be output if the total number (MsgCnt) of messages in the message queue is not 0.

Special Features

- With a request, only one message is requested, regardless of the total number of messages in the message queue.
- By specifying a message type, a specific message can be requested.

Reading Messages

General

Example: A RegAcc sends a QUEUE-READ request to the RRI.

- There are 3 messages in the message queue,
  - 2 of type “expireWarning”,
  - 1 of type “authInfoExpire”.
- The “authInfoExpire” message is the message with the oldest date.

Result: The message with the oldest date, i.e. “authInfoExpire”, is output.

Specific

A specific message type can be issued by specifying the keyword “MsgType” followed by the message type, e.g. “expireWarning”. If there are several messages of a type in the message queue, the message with the oldest date is issued.

Automatic Reminders for Older Messages

- If messages in the message queue are older than three days, the RegAcc is automatically informed. This is done by sending an e-mail to the “EMailTo” address from the RegAcc profile.
- The notification e-mail is sent once a day if the messages remain in the message queue (and are older than three days).

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	queue-read	1	enumeration	queuereaderule	request type
Version	-	1	enumeration	versionerule	version, only relevant for requests in key/value format
MsgType	msgtype	0-1	enumeration	msgtyperule	type of message; if no MsgType is specified, the oldest message (sorted by MsgTime) is displayed

QUEUE-READ Responses

Tables for the QUEUE-READ Responses

A response from the message queue is sent for every QUEUE-READ job. Some lines in the responses differ depending on the message type. In the tables for the various message types, the lines for the data fields that only apply to the message type are colored orange.

The tables for the message types are:

- contactDelete
- authInfo2Delete
- authInfo2Notify
- authInfoExpire
- chprovAuthInfo
- domainDelete
- expireWarning and expire (commoin table)

Special Features

- The messages contain the information “MsgTime”, which stores the times of message creation.
- DENIC assigns a unique ID, the so-called “MsgId”, which is used to delete messages.
- The message with the oldest date is output.
- A counter (MsgCnt) is also output with each message, which still indicates all existing messages in the message queue.

QUEUE-DELETE Request

Function

The QUEUE-DELETE request is used to delete messages from the RRI message queue.

Features

You can only delete the oldest message of a message type. You must always state "Msgld".

Request Parameters

A request is composed of the fields described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Action	queue-delete	1	enumeration	queuer-delete-erule	This is the request type.
Version	-	1	enumeration	version-erule	This is the version. The keyword is only relevant for requests in the key/-value format.
CtId	ctid	0 - 1	token 3 - 64	Any visible Unicode character (according to Unicode version 3.1)	This is the unique transaction ID which is issued by the client.
MsgId	msgid	1	token 30-255	msgid-rule	This is the unique message ID.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgType	msgtype	0-1	enumeration	msgtype-erule	This is the message type; if the keyword "MsgType" contains an empty value, the oldest message (sorted according to MsgTime) will be deleted.

Common Mistakes

A QUEUE-DELETE request with an "MsgId" is tried to be submitted; but according to "MsgTime" the message referenced by the ID is not the oldest one ("MsgTime" is output as part of the response to a QUEUE-READ request).

QUEUE-DELETE Response

Description of the Data Fields in QUEUE-DELETE

The response of the .de registration system to a QUEUE-DELETE request comprises the data fields described below:

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Processing result success = order was carried out failed = order was not carried out exception = system malfunction, order was not carried out
StId	stid	1	token 30-255	stid-rule	This is a Server Transaction ID (STID) which is assigned by the server. The Stid ensures that the request from the provider and DENIC's response are referenced with a globally unique ID.

Error Messages Related to QUEUE-READ and QUEUE-DELETE Requests

Potential Error Messages

Below you find a list of the error messages that may occur exclusively in relation with the processing of QUEUE-READ or QUEUE-DELETE requests.

Comments on the Table

- The “Input Format” column indicates whether the error message can occur when using the key/value format (= KV) and/or XML format (= XML).
- The “Read / Delete” column indicates whether the error message can occur when reading (= R) and / or deleting (= D) a message.

Error Code	Error Message	Input Format	Read / Delete	Brief Description
63600063502	Inadmissible value for keyword "MsgType"	KV / XML	R/D	The specified MsgType does not exist.
63650063607	This message is not the oldest	KV / XML	D	Only the oldest message in the queue can be deleted at any one time; however, the specified message is not the oldest.
63650063609	MsgId does not exist	KV / XML	D	The specified MsgId does not exist.
63650043610	Keyword "MsgId" cannot be recognized	KV / XML	D	The keyword “MsgId” is missing or cannot be recognized.

For orders in XML format, a schema validation error occurs if a “MsgId” is not recognized, see chapter Description of Data Formats (Key/Value / XML), section "Validation Errors in XML Requests".

authInfo2Delete Message Type

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of processing success = order was carried out failed = order was not carried out exception = system malfunction, request was not carried out
StId	stid	1	token 30-255	stid-rule	A server transaction ID (StId) assigned by the server. The StId is used to globally and unambiguously reference the provider's request and the corresponding response from DENIC.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgCnt	msgcnt	1	token 1 - 7	digit-rule	Without “MsgType”, the total number of messages is displayed; with “MsgType”, the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.
MsgTime	msgtime	1	dateTime 25	date-rule	The time of message creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgType	-	1	enumeration	msgtype-erule	The category of the message issued
Domain	domain:handle	1	token 4 – 66	domain-rule	Domain name
Domain-Ace	domain:ace	0-1	token 4 - 66	domainace-rule	Domain name in ACE form (ASCII Compatible Encoding).
Message	<msg:message level="info" code="11300077725"> <tr:text>AuthInfo deleted</tr:text> </msg:message>	1	normalizedString 1 - 255	message-rule	First message line with a fixed code and text.
Message	<msg:message level="info" code="11300077726"> <tr:text>AuthInfo letter could not be delivered</tr:text> </msg:message>	1	normalizedString 1 - 255	message-rule	Second message line with a fixed code and text.

authInfo2Notify Message Type

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of processing success = order was carried out failed = order was not carried out exception = system malfunction, request was not carried out
StId	stid	1	token 30-255	stid-rule	A server transaction ID (StId) assigned by the server. The StId is used to globally and unambiguously reference the provider's request and the corresponding response from DENIC.
MsgCnt	msgcnt	1	token 1 - 7	digit-rule	Without “MsgType”, the total number of messages is displayed; with “MsgType”, the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgTime	msgtime	1	dateTime 25	date-rule	The time of message creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message
MsgType	-	1	enumeration	msgtype-erule	The category of the message issued
Domain	domain:handle	1	token 4 – 66	domain-rule	Domain name
Domain-Ace	domain:ace	0-1	token 4 - 66	domainace-rule	Domain name in ACE form (ASCII Compatible Encoding).

authInfoExpire Message Type

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of processing success = order was carried out failed = order was not carried out exception = system malfunction, request was not carried out
StId	stid	1	token 30-255	stid-rule	A server transaction ID (StId) assigned by the server. The StId is used to globally and unambiguously reference the provider's request and the corresponding response from DENIC.

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgCnt	msgcnt	1	token 1 - 7	digit-rule	Without “MsgType”, the total number of messages is displayed; with “MsgType”, the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.
MsgTime	msgtime	1	dateTime 25	date-rule	The time of message creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message
MsgType	-	1	enumeration	msgtype-erule	The category of the message issued
Domain	domain:handle	1	token 4 – 66	domain-rule	Domain name

K/V Keyword	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Domain-Ace	domain:ace	0-1	token 4 - 66	domainace-rule	Domain name in ACE form (ASCII Compatible Encoding).
AuthInfoExpire	authinfoexpire	0-1	dateTime 25	date-rule	AuthInfo expiration date

chprovAuthInfo Message Type

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of processing success = order was carried out failed = order was not carried out exception = system malfunction, request was not carried out

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
StId	stid	1	token 30-255	stid-rule	A server transaction ID (StId) assigned by the server. The StId is used to globally and unambiguously reference the provider's request and the corresponding response from DENIC.
MsgCnt	msgcnt	1	token 1 - 7	digit-rule	Without “MsgType”, the total number of messages is displayed; with “MsgType”, the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.
MsgTime	msgtime	1	dateTime 25	date-rule	The time of message creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgType	-	1	enumeration	msgtype-erule	The category of the message issued
Domain	domain:handle	1	token 4 – 66	domain-rule	Domain name
Domain-Ace	domain:ace	0-1	token 4 - 66	domainace-rule	Domain name in ACE form (ASCII Compatible Encoding).
Chprov-To	new	0-1	token 7-14	msgid-rule	RegAccId of the RegAcc that started the CHPROV.
Chprov-From	old	0-1	token 7-14	msgid-rule	RegAccId of the RegAcc that administered the domain.

contactDelete Message Type

Special Features

- Several deleted DENIC handles (deleted contacts) can be grouped together in a “contactDelete” response.
- A maximum of 100 DENIC handles can be listed in a “contactdelete” response.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
RESULT	result	1	enumeration	result-erule	Result of processing success = order was carried out failed = order was not carried out exception = system malfunction, request was not carried out
STID	stid	1	token 30-255	stid-rule	A server transaction ID (StId) assigned by the server. The StId is used to globally and unambiguously reference the provider's request and the corresponding response from DENIC.
msgcnt	msgcnt	1	token 1 - 7	digit-rule	Without “MsgType”, the total number of messages is displayed; with “MsgType”, the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
msgtime	msgtime	1	dateTime 25	date-rule	The time of message creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message
MsgType	-	1	enumeration	msgtype-erule	The category of the message issued
handle	<msg:handle> [Handle-ID] </msg:handle>	1-100	token 9 - 32	message-rule	• A handle ID is specified in each line (K/V) or each element (XML). • A maximum of 100 handle IDs can be in a response.

contactVerificationRequired Message Type

The RegAcc is informed with this message type that

- a verification is required for a contact,
- by when the contact must be verified, and what

- will happen in the event of failure to comply.

> **Notice**
>
> Several domains of a RegAcc may be affected by deadlines. However, only one domain is specified in the message. This is a domain with the earliest deadlines.

Examples

- xml
- k/v
- e-mail

format: xml

request: contactVerificationRequired

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:msg="http://registry.denic.de/msg/5.0">
  <msg:queue-read msgType="contactVerificationRequired"/>
</registry-request>
```

format: xml

response: contactVerificationRequired

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="h-
ttp://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>e5b905e9-f867-4fcd-8e93-e13db4ff3065</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
<msg:message xmlns:msg="http://registry.denic.de/msg/5.0" msgid="8960348c-6879-cb92-2b6f-
cbc9abf98080" msgcnt="1" msgtime="2024-06-01T15:51:08+02:00">
  <msg:contactVerificationRequired>
    <msg:handle>DENIC-99995-BAD-GUY</msg:handle>
    <msg:verificationClaims>
      <msg:claim>name</msg:claim>
      <msg:claim>address</msg:claim>
    </msg:verificationClaims>
    <msg:numberOfReferencedDomainsTotal>2753</msg:numberOfReferencedDomainsTotal>
    <msg:earliestDomainForDedelegation>
      <msg:domain>
        <msg:handle>denic.de</msg:handle>
        <msg:ace>denic.de</msg:ace>
      </msg:domain>
```

```
      <msg:verificationDeadline>2024-06-06T15:51:08+02:00</msg:verificationDeadline>
      </msg:earliestDomainForDedelegation>
    <msg:earliestDomainForDeletion>
       <msg:domain>
         <msg:handle>denic.de</msg:handle>
         <msg:ace>denic.de</msg:ace>
       </msg:domain>
      <msg:verificationDeadline>2024-06-13T15:51:08+02:00</msg:verificationDeadline>
    </msg:earliestDomainForDeletion>
    <msg:message level="info" code="16350000050">
      <tr:text>Verification information must be provided for the handle to avoid dedelegation and/or dele-
tion</tr:text>
    </msg:message>`,
    <msg:message level="info" code="16350000051">
      <tr:text>Please provide the requested verification information by means of a Contact UPDATE</tr:text>
    </msg:message>
  </msg:contactVerificationRequired>
</msg:message>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

request: contactVerificationRequired

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:msg="http://registry.denic.de/msg/5.0">
  <msg:delete msgid="02699380-9c7a-11de-8188-9ff72af94906" msgType="contactVerificationRequired"/>
</registry-request>
```

format: xml

general response to all QUEUE-DELETE requests

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>078db1bf-9f0a-11df-a9e2-9519b5688882</tr:stid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: k/v

request: contactVerificationRequired

```
Version: 5.0
Action: queue-read
Msgtype: contactdelete
```

format: k/v

response: contactVerificationRequired

```
Result: success
STID: 253526e9-9f04-11df-a9e2-9519b5688882
Msgcnt: 1
Msgtime: 2024-06-01T15:51:08+02:00
Msgid: 8960348c-6879-cb92-2b6f-cbc9abf98080
msgtype: contactVerificationRequired
Handle: DENIC-99995-BAD-GUY
VerificationClaim: name
VerificationClaim: address
NumberOfReferencedDomainsTotal: 2753
Message: 16350000050 Verification information must be provided for the handle to avoid dedelegation and/or
deletion
Message: 16350000051 Please provide the requested verification information by means of a Contact UPDATE

[earliest domain for dedelegation]
Domain: denic.de
```

```
Domain-ace: denic.de
VerificationDeadlineBeforeDedelegation: 2024-06-06T15:51:08+02:00

[earliest domain for deletion]
Domain: denic.de
Domain-ace: denic.de
VerificationDeadlineBeforeDeletion: 2024-06-13T15:51:08+02:00
```

format: k/v

request: contactVerificationRequired

```
Version: 5.0
Action: queue-delete
Msgid: b988dae0-ae64-11de-b15c-f1279ab97beb
Msgtype: contactVerificationRequired
CTID: bre-34634621634
```

format: k/v

general response to all QUEUE-DELETE requests

```
Result: success
```

```
2 | STID: 0b97f9b7-9f0b-11df-a9e2-9519b5688882
3 | CTID: bre-34634621634
```

format: e-mail

message type: contactVerificationRequired

environment: live

```
 1 | From:    registry-response@denic.de
 2 | To:      <regacc-email>
 3 | Subject: CONTACT VERIFICATION REQUIRED - $date$
 4 |
 5 | STATUS:
 6 | TICKET:
 7 | OBJECT: Contact
 8 | HANDLE: DENIC-99995-BAD-GUY
 9 | INFO: 16350000052 Claims to verify for this handle [name, address]
10 | INFO: 16350000053 Number of referenced domains for this handle [2753]
11 | INFO: 16350000054 Earliest domain for dedelegation [denic.de, <verification-deadline-before-dedelegation>]
12 | INFO: 16350000055 Earliest domain for deletion [denic.de, <verification-deadline-before-deletion>]
13 | INFO: 16350000050 Verification information must be provided for the handle to avoid dedelegation and/or
   | deletion
14 | INFO: 16350000051 Please provide the requested verification information by means of a Contact UPDATE
```

format: e-mail

message type: contactVerificationRequired

environment: test

```
From:    registry-response@test.denic.de
To:      <regacc-email>
Subject: contactVerificationRequired TEST - $date$

STATUS:
TICKET:
OBJECT: Contact
HANDLE: DENIC-99995-BAD-GUY
INFO: 16350000052 Claims to verify for this handle [name, address]
INFO: 16350000053 Number of referenced domains for this handle [2753]
INFO: 16350000054 Earliest domain for dedelegation [denic.de, <verification-deadline-before-dedelegation>]
INFO: 16350000055 Earliest domain for deletion [denic.de, <verification-deadline-before-deletion>]
INFO: 16350000050 Verification information must be provided for the handle to avoid dedelegation and/or
deletion
INFO: 16350000051 Please provide the requested verification information by means of a Contact UPDATE
```

domainDelete Message Type

Further content in this topic

- "Special features of the domainDelete message type" on page 490
- "Examples" on page 492

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of pro-cessing success = order was carried out failed = order was not carried out exception = sys-tem malfunction, request was not carried out

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
StId	stid	1	token 30-255	stid-rule	A server trans-action ID (StId) assigned by the server. The StId is used to glob-ally and unam-biguously reference the provider's request and the corresponding response from DENIC.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgCnt	msgcnt	1	token 1 - 7	digit-rule	Without “MsgType”, the total number of messages is displayed; with “MsgType”, the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgTime	msgtime	1	dateTime 25	date-rule	The time of mes-sage creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message
MsgType	-	1	enumeration	msgtype-erule	The category of the message issued
Domain	domain:handle	1	token 4 – 66	domain-rule	Domain name
Domain-Ace	domain:ace	0-1	token 4 - 66	domainace-rule	Domain name in ACE form (ASCII Compatible Encoding).

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Message	<msg:message level="info" code-e="16350000031„> <tr:text>Domain has been deleted [request-reference DENIC#:[Tick-etnummer]</tr:text></msg:message>	1	normalizedString 1 - 255	message-rule	Message line with a fixed code, text and a vari-able ticket num-ber Note: The ticket number is a reference to a process and must be provided to the DENIC legal department in the event of quer-ies.

Special features of the domainDelete message type

- "domainDelete" is a message about a domain deletion carried out by DENIC.
- Their could be different reasons in the information part of the message, why a domain was deleted. One of them could be reference to a ticket ID from DENIC's internal ticket management system.

- In case of a ticket ID or no ticket ID/no reason our member support DENIC Business Services (DBS) can help with information.

- In a K/V format example, the message lines would look like this:
  - **Domain in “serverHold” status, the verification period has expired:**

```
message: 16350000031 Domain has been deleted [no verification information provided]
```

  - **Domain with a ticket ID:**

```
message: 16350000031 Domain has been deleted [request-reference DENIC#:2024100143000000]
```

  - **Domain without ticket ID and without reason for deletion:**

```
message: 16350000031 Domain has been deleted []
```

Examples

- xml
- k/v
- e-mail

format: xml

request: domainDelete

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:msg="http://registry.denic.de/msg/5.0">
  <msg:queue-read msgType="domainDelete"/>
</registry-request>
```

format: xml

response: domainDelete

```
<?xml version="1.0" encoding="UTF-8"?>

```

```
<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>ae029b0a-50af-11e0-b375-95a6041ce317</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
      <msg:message xmlns:msg="http://registry.denic.de/msg/5.0" msgid="ee6bfa7f-4afd-11e0-ad4b-47ecb35e56d2" msgcnt="1" msgtime="2011-03-10T11:05:30+01:00">
        <msg:domainDelete>
          <msg:domain>
            <msg:handle>de-domain.de</msg:handle>
            <msg:ace>de-domain.de</msg:ace>
          </msg:domain>
          <msg:message level="info" code="16350000031">
            <tr:text>Domain has been deleted</tr:text>
            <tr:argument>request-reference DENIC#:20110310430000000</tr:argument>
          </msg:message>
        </msg:domainDelete>
      </msg:message>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

response: domainDelete

status: was serverHold

verification time up

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>ae029b0a-50af-11e0-b375-95a6041ce317</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
<msg:message xmlns:msg="http://registry.denic.de/msg/5.0" msgid="8960348c-6879-cb92-2b6f-cbc9abf98080" msgcnt="1" msgtime="2024-06-01T15:51:08+02:00">
  <msg:domainDelete>
    <msg:domain>
      <msg:handle>de-example.de</msg:handle>
      <msg:ace>de-example.de</msg:ace>
    </msg:domain>
    <msg:message level="info" code="16350000031">
      <tr:text>Domain has been deleted</tr:text>
      <tr:argument>no verification information provided</tr:argument>
    </msg:message>
  </msg:domainDelete>
</msg:message>
    </tr:data>
```

format: xml**response:** domainDelete**status:** was serverHold**verification** time up

```
  </tr:transaction>
</registry-response>
```

format: xml

response: domainDelete

ticket: available

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="h-
ttp://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>ae029b0a-50af-11e0-b375-95a6041ce317</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
<msg:message xmlns:msg="http://registry.denic.de/msg/5.0" msgid="8960348c-6879-cb92-2b6f-
cbc9abf98080" msgcnt="1" msgtime="2024-06-01T15:51:08+02:00">
  <msg:domainDelete>
    <msg:domain>
      <msg:handle>de-example.de</msg:handle>
      <msg:ace>de-example.de</msg:ace>
```

format: xml**response:** domainDelete**ticket:** available

```
    </msg:domain>
    <msg:message level="info" code="16350000031">
      <tr:text>Domain has been deleted</tr:text>
      <tr:argument>request-reference DENIC#:2024100143000000</tr:argument>
    </msg:message>
  </msg:domainDelete>
</msg:message>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

request: domainDelete

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:msg="http://registry.denic.de/msg/5.0">
  <msg:delete msgid="02699380-9c7a-11de-8188-9ff72af94906" msgType="domainDelete"/>
</registry-request>
```

format: xml

general response to all QUEUE-DELETE requests

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>078db1bf-9f0a-11df-a9e2-9519b5688882</tr:stid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainDelete

```
Version: 5.0
Action: queue-read
Msgtype: domaindelete
```

format: k/v

response: domainDelete

```
Result: success
STID: 78bf488c-9f0c-11df-a9e2-9519b5688882
Msgcnt: 1
Msgtime: 2011-03-10T11:05:30+01:00
Msgid: ee6bfa7f-4afd-11e0-ad4b-47ecb35e56d2
Msgtype: domainDelete
Domain: de-example.de
Domain-ace: de-example.de
Message: 16350000031 Domain has been deleted []
```

format: k/v

response: domainDelete

status: was serverHold

verification time up

```
Result: success
STID: 253526e9-9f04-11df-a9e2-9519b5688882
Msgcnt: 1
```

```
Msgtime: 2024-06-01T15:51:08+02:00
Msgid: 8960348c-6879-cb92-2b6f-cbc9abf98080
Msgtype: domainDelete
Domain: de-example.de
Domain-ace: de-example.de
Message: 16350000031 Domain has been deleted [no verification information provided]
```

format: k/v

response: domainDelete

ticket: available

```
Result: success
STID: 253526e9-9f04-11df-a9e2-9519b5688882
Msgcnt: 1
Msgtime: 2024-06-01T15:51:08+02:00
Msgid: 8960348c-6879-cb92-2b6f-cbc9abf98080
Msgtype: domainDelete
Domain: de-example.de
Domain-ace: de-example.de
Message: 16350000031 Domain has been deleted [request-reference DENIC#:2024100143000000]
```

format: k/v

request: domainDelete

```
Version: 5.0
Action: queue-delete
Msgid: b988dae0-ae64-11de-b15c-f1279ab97beb
Msgtype: domainDelete
CTID: bre-34634621634
```

format: k/v

general response to all QUEUE-DELETE requests

```
Result: success
STID: 0b97f9b7-9f0b-11df-a9e2-9519b5688882
CTID: bre-34634621634
```

format: e-mail

message type: domainDelete

environment: live

```
From: registry-response@denic.de
```

```
Subject: DOMAIN DELETE information - 2023-09-27T16:15:15+02:00

STATUS: success
TICKET:
OBJECT: Domain
HANDLE: dänic.de [xn--dnic-loa.de]
INFO:   16350000031 Domain has been deleted
```

format: e-mail

message type: domainDelete

environment: live

status: was serverHold

verification time up

```
From:    registry-response@denic.de
To:      <regacc-email>
Subject: DOMAIN DELETE information - 2023-09-27T16:15:15+02:00

STATUS: success
TICKET:
OBJECT: Domain
```

```
HANDLE: dänic.de [xn--dnic-loa.de]
INFO:   16350000031 Domain has been deleted [no verification information provided]
```

format: e-mail

message type: domainDelete

environment: live

ticket: available

```
From:    registry-response@test.denic.de
To:      <regacc-email>
Subject: DOMAIN DELETE information TEST - 2023-09-27T16:15:15+02:00

STATUS: success
TICKET: DENIC#:2024100143000000
OBJECT: Domain
HANDLE: dänic.de [xn--dnic-loa.de]
INFO:   16350000031 Domain has been deleted
```

format: e-mail

message type: domainDelete

environment: test

```
From: registry-response@test.denic.de
Subject: DOMAIN DELETE information TEST - 2023-09-27T16:15:15+02:00

STATUS: success
TICKET:
OBJECT: Domain
HANDLE: dänic.de [xn--dnic-loa.de]
INFO:   16350000031 Domain has been deleted
```

format: e-mail

message type: domainDelete

environment: test

status: was serverHold

verification time up

```
From:    registry-response@test.denic.de
```

```
To:      <regacc-email>
Subject: DOMAIN DELETE information TEST - 2023-09-27T16:15:15+02:00

STATUS: success
TICKET:
OBJECT: Domain
HANDLE: dänic.de [xn--dnic-loa.de]
INFO:   16350000031 Domain has been deleted [no verification information provided]
```

format: e-mail

message type: domainDelete

environment: test

ticket: available

```
From:    registry-response@test.denic.de
To:      <regacc-email>
Subject: DOMAIN DELETE information TEST - 2023-09-27T16:15:15+02:00

STATUS: success
TICKET: DENIC#:2024100143000000
OBJECT: Domain
HANDLE: dänic.de [xn--dnic-loa.de]
```

```
9 | INFO:   16350000031 Domain has been deleted
```

domainStatusUpdate Message Type

- The message type indicates a status for a domain and is generated after a successful request when:
    - a domain has been registered,
    - a domain's data has been changed, or
    - a domain has been connected.
    - Additional verification information may be required based on the risk assessment.
        - If verification information is required, the message contains additional information.
- If there are multiple domain holders for a domain, a so-called multi-holder domain, all domain holders are listed for whom no verification information is required.
- A domain in quarantine must be verified, and the domain is also de-delegated, which is indicated by the “serverHold” status.

Examples

- xml
- k/v

- e-mail

format: xml

request: domainStatusUpdate

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:msg="http://registry.denic.de/msg/5.0">
  <msg:queue-read msgType="domainStatusUpdate"/>
</registry-request>
```

format: xml

response: domainStatusUpdate

status: connect

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="h-
ttp://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>ae029b0a-50af-11e0-b375-95a6041ce317</tr:stid>
    <tr:result>success</tr:result>
```

format: xml**response:** domainStatusUpdate**status:** connect

```
    <tr:data>
<msg:message xmlns:msg="http://registry.denic.de/msg/5.0" msgid="8960348c-6879-cb92-2b6f-cbc9abf91616" msgcnt="1" msgtime="2024-06-01T15:51:08+02:00">
  <msg:domainStatusUpdate>
    <msg:domain>
      <msg:handle>de-example.de</msg:handle>
      <msg:ace>de-example.de</msg:ace>
    </msg:domain>
    <msg:holders>
      <msg:handle>DENIC-1000002-GoodGuy</msg:handle>
      <msg:handle>DENIC-1000002-GoodGuy2</msg:handle>
    </msg:holders>
    <msg:status>connect</msg:status>
  </msg:domainStatusUpdate>
</msg:message>
    </tr:data>
  </tr:transaction>
</registry-response>

```

format: xml

response: domainStatusUpdate

status: connect

verification: required

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>ae029b0a-50af-11e0-b375-95a6041ce317</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
<msg:message xmlns:msg="http://registry.denic.de/msg/5.0" msgid="8960348c-6879-cb92-2b6f-cbc9abf91616" msgcnt="1" msgtime="2024-06-01T15:51:08+02:00">
  <msg:domainStatusUpdate>
    <msg:domain>
      <msg:handle>de-example.de</msg:handle>
      <msg:ace>de-example.de</msg:ace>
    </msg:domain>
    <msg:holders>
      <msg:handle>DENIC-1000002-MaybeGoodGuy</msg:handle>
      <msg:handle>DENIC-1000002-GoodGuy</msg:handle>
    </msg:holders>
    <msg:status>connect</msg:status>
    <msg:verificationDeadlineBeforeDedelegation>2024-06-06T15:51:08+02:00</msg:verificationDeadlineBeforeDedelegation>
```

format: xml**response:** domainStatusUpdate**status:** connect**verification:** required

```
    <msg:verificationDeadlineBeforeDeletion>2024-06-13T15:51:08+02:00</ms-
g:verificationDeadlineBeforeDeletion>
    <msg:message level="info" code="16350000040">
      <tr:text>Verification information must be provided for the holder(s) to avoid dedelegation by</tr:-
text>
        <tr:argument>Date: 2024-06-06T15:51:08+02:00</tr:argument>
        <tr:argument>VerificationClaims: address;name</tr:argument>
     </msg:message>
    <msg:verificationDeadlineBeforeDeletion>2024-06-13T15:51:08+02:00</ms-
g:verificationDeadlineBeforeDeletion>
    <msg:message level="info" code="16350000041">
      <tr:text>Verification information must be provided for the holder(s) to avoid deletion by</tr:text>
      <tr:argument>Date: 2024-06-13T15:51:08+02:00</tr:argument>
      <tr:argument>VerificationClaims: address;name</tr:argument>
    </msg:message>
   </msg:domainStatusUpdate>
</msg:message>
    </tr:data>
  </tr:transaction>
</registry-response>
```

format: xml

response: domainStatusUpdate

status: serverHold

verification: required

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>ae029b0a-50af-11e0-b375-95a6041ce317</tr:stid>
    <tr:result>success</tr:result>
    <tr:data>
<msg:message xmlns:msg="http://registry.denic.de/msg/5.0" msgid="8960348c-6879-cb92-2b6f-cbc9abf91616" msgcnt="1" msgtime="2024-06-01T15:51:08+02:00">
  <msg:domainStatusUpdate>
    <msg:domain>
      <msg:handle>de-example.de</msg:handle>
      <msg:ace>de-example.de</msg:ace>
    </msg:domain>
    <msg:holders>
      <msg:handle>DENIC-1000002-BadGuy</msg:handle>
      <msg:handle>DENIC-1000002-GoodGuy</msg:handle>
    </msg:holders>
    <msg:status>serverHold</msg:status>
    <msg:verificationDeadlineBeforeDeletion>2024-06-13T15:51:08+02:00</msg:verificationDeadlineBeforeDeletion>
```

format: xml**response:** domainStatusUpdate**status:** serverHold**verification:** required

```
20      <msg:message level="info" code="16350000041">
21        <tr:text>Verification information must be provided for the holder(s) to avoid deletion by</tr:text>
22        <tr:argument>Date: 2024-06-13T15:51:08+02:00</tr:argument>
23        <tr:argument>VerificationClaims: address;name</tr:argument>
24      </msg:message>
25    </msg:domainStatusUpdate>
26  </msg:message>
27      </tr:data>
28    </tr:transaction>
29  </registry-response>
```

format: xml

request: domainStatusUpdate

```
1  <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
2
3  <registry-request xmlns="http://registry.denic.de/global/5.0" xmlns:msg="http://registry.denic.de/msg/5.0">
4    <msg:delete msgid="02699380-9c7a-11de-8188-9ff72af94906" msgType="domainStatusUpdate"/>
5  </registry-request>
```

format: xml

general response to all QUEUE-DELETE requests

```xml
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/5.0" xmlns:tr="http://registry.denic.de/transaction/5.0">
  <tr:transaction>
    <tr:stid>078db1bf-9f0a-11df-a9e2-9519b5688882</tr:stid>
    <tr:result>success</tr:result>
  </tr:transaction>
</registry-response>
```

format: k/v

request: domainStatusUpdate

```
Version: 5.0
Action: queue-read
Msgtype: domainstatusupdate
```

format: k/v

response: domainStatusUpdate

status: connect

```
Result: success
STID: 78bf488c-9f0c-11df-a9e2-9519b5688882
Msgcnt: 1
Msgtime: 2024-06-01T15:51:08+02:00
Msgid: 8960348c-6879-cb92-2b6f-cbc9abf91616
Msgtype: domainStatusUpdate
Domain: de-example.de
Domain-ace: de-example.de
Holder: DENIC-1000002-GoodGuy
Holder: DENIC-1000002-GoodGuy2
Status: connect
```

format: k/v

response: domainStatusUpdate

status: connect

verification: required

```
Result: success
STID: 78bf488c-9f0c-11df-a9e2-9519b5688882
Msgcnt: 1
Msgtime: 2024-06-01T15:51:08+02:00
Msgid: 8960348c-6879-cb92-2b6f-cbc9abf91616
Msgtype: domainStatusUpdate
Domain: de-example.de
Domain-ace: de-example.de
Holder: DENIC-1000002-MaybeGoodGuy
Holder: DENIC-1000002-GoodGuy
Status: connect
VerificationDeadlineBeforeDedelegation: 2024-06-06T15:51:08+02:00
VerificationDeadlineBeforeDeletion: 2024-06-13T15:51:08+02:00
Message: 16350000040 Verification information must be provided for the holder(s) to avoid dedelegation by [Date: 2024-06-06T15:51:08+02:00, VerificationClaims: address;name]
Message: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [Date: 2024-06-13T15:51:08+02:00, VerificationClaims: address;name]
```

format: k/v

response: domainStatusUpdate

status: serverHold

verification: required

```
Result: success
STID: 78bf488c-9f0c-11df-a9e2-9519b5688882
Msgcnt: 1
Msgtime: 2024-06-01T15:51:08+02:00
Msgid: 8960348c-6879-cb92-2b6f-cbc9abf91616
Msgtype: domainStatusUpdate
Domain: de-example.de
Domain-ace: de-example.de
Holder: DENIC-1000002-BadGuy
Holder: DENIC-1000002-GoodGuy
Status: serverHold
VerificationDeadlineBeforeDeletion: 2024-06-13T15:51:08+02:00
Message: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [Date:
2024-06-13T15:51:08+02:00, VerificationClaims: address;name]
```

format: k/v

request: domainStatusUpdate

```
Version: 5.0
```

```
Action: queue-delete
Msgid: b988dae0-ae64-11de-b15c-f1279ab97beb
Msgtype: domainDelete
CTID: bre-34634621634
```

format: k/v

general response to all QUEUE-DELETE requests

```
Result: success
STID: 0b97f9b7-9f0b-11df-a9e2-9519b5688882
CTID: bre-34634621634
```

format: e-mail

message type: domainStatusUpdate

environment: live

status: connect

```
From:    registry-response@denic.de
To:      <regacc-email>
Subject: DOMAIN STATUS UPDATE - $date$
```

```
 4
 5  STATUS: success
 6  TICKET:
 7  OBJECT: Domain
 8  HANDLE: dänic.de [xn--dnic-loa.de]
 9  HOLDER: DENIC-1000002-MaybeGoodGuy
10  HOLDER: DENIC-1000002-GoodGuy
11  INFO: 53000080013 Domain "Status" is "connect"
```

format: e-mail

message type: domainStatusUpdate

environment: live

status: connect

verification: required

```
1  From:    registry-response@denic.de
2  To:      <regacc-email>
3  Subject: DOMAIN STATUS UPDATE - $date$
4
5  STATUS: success
6  TICKET:
```

```
 7 | OBJECT: Domain
 8 | HANDLE: dänic.de [xn--dnic-loa.de]
 9 | HOLDER DENIC-1000002-MaybeGoodGuy
10 | HOLDER DENIC-1000002-GoodGuy
11 | INFO: 53000080013 Domain "Status" is "connect"
12 | INFO: 16350000040 Verification information must be provided for the holder(s) to avoid dedelegation by
   | [Date: 2025-06-13T15:51:08+02:00, VerificationClaims: address;name]
13 | INFO: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [Date:
   | 2025-06-13T15:51:08+02:00, VerificationClaims: address;name]
```

format: e-mail

message type: domainStatusUpdate

environment: live

status: serverHold

verification: required

```
1 | From:    registry-response@denic.de
2 | To:      <regacc-email>
3 | Subject: DOMAIN STATUS UPDATE - $date$
4 |
5 | STATUS: success
```

```
 6 | TICKET:
 7 | OBJECT: Domain
 8 | HANDLE: dänic.de [xn--dnic-loa.de]
 9 | HOLDER DENIC-1000002-MaybeGoodGuy
10 | HOLDER DENIC-1000002-GoodGuy
11 | INFO: 53000080015 Domain "Status" is "serverHold"
12 | INFO: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [Date:
   | 2025-06-13T15:51:08+02:00, VerificationClaims: address;name]
```

format: e-mail

message type: domainStatusUpdate

environment: test

status: connect

```
1 | From:    registry-response@test.denic.de
2 | To:      <regacc-email>
3 | Subject: DOMAIN STATUS UPDATE TEST - $date$
4 |
5 | STATUS: success
6 | TICKET:
7 | OBJECT: Domain
8 | HANDLE: dänic.de [xn--dnic-loa.de]
9 | HOLDER: DENIC-1000002-MaybeGoodGuy
```

```
10 | HOLDER: DENIC-1000002-GoodGuy
11 | INFO: 53000080013 Domain "Status" is "connect"
```

format: e-mail

message type: domainStatusUpdate

environment: test

status: connect

verification: required

```
 1 | From:    registry-response@test.denic.de
 2 | To:      <regacc-email>
 3 | Subject: DOMAIN STATUS UPDATE TEST - $date$
 4 |
 5 | STATUS: success
 6 | TICKET:
 7 | OBJECT: Domain
 8 | HANDLE: dänic.de [xn--dnic-loa.de]
 9 | HOLDER DENIC-1000002-MaybeGoodGuy
10 | HOLDER DENIC-1000002-GoodGuy
11 | INFO: 53000080013 Domain "Status" is "connect"
```

```
12 | INFO: 16350000040 Verification information must be provided for the holder(s) to avoid dedelegation by [ate:
   | 2025-06-06T15:51:08+02:00, VerificationClaims: address;name]
13 | INFO: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [Date:
   | 2025-06-13T15:51:08+02:00, VerificationClaims: address;name]
```

format: e-mail

message type: domainStatusUpdate

environment: test

status: serverHold

verification: required

```
 1 | From:    registry-response@test.denic.de
 2 | To:      <regacc-email>
 3 | Subject: DOMAIN STATUS UPDATE TEST - $date$
 4 |
 5 | STATUS: success
 6 | TICKET:
 7 | OBJECT: Domain
 8 | HANDLE: dänic.de [xn--dnic-loa.de]
 9 | HOLDER DENIC-1000002-MaybeGoodGuy
10 | HOLDER DENIC-1000002-GoodGuy
```

```
11 | INFO: 53000080015 Domain "Status" is "serverHold"
12 | INFO: 16350000041 Verification information must be provided for the holder(s) to avoid deletion by [Date:
   | 2025-06-13T15:51:08+02:00, VerificationClaims: address;name]
```

expireWarning and expire Message Type

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Result	result	1	enumeration	result-erule	Result of processing success = order was carried out failed = order was not carried out exception = system malfunction, request was not carried out
StId	stid	1	token 30-255	stid-rule	A server transaction ID (StId) assigned by the server. The StId is used to globally and unambiguously reference the provider's request and the corresponding response from DENIC.

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
MsgCnt	msgcnt	1	token 1 - 7	digit-rule	Without “MsgType”, the total number of messages is displayed; with “MsgType”, the number of messages for the specific type is displayed. For K/V format only: if this field contains the value 0, there are no messages in the entire message queue. For XML, msgcnt is only displayed if there is at least one message in the queue.
MsgTime	msgtime	1	dateTime 25	date-rule	The time of message creation
MsgId	msgid	1	token 30-255	msgid-rule	The message ID of the message
MsgType	-	1	enumeration	msgtype-erule	Two different message types: ExpireWarning comes one week before the domain expires. Expire comes on the day the domain expired.
Domain	domain:handle	1	token 4 – 66	domain-rule	Domain name

K/V Key-word	XML Namespace and Element	Occurrence min - max	Type / Length	Value Range	Description
Domain-Ace	domain:ace	0-1	token 4 - 66	domainace-rule	Domain name in ACE form (ASCII Compatible Encoding).
Expire	expire	0-1	dateTime 25	date-rule	Domain expiration date

Sub-Title1

07 Message Codes

Message Codes 526
Structure and Components of Message Codes 526
Parsing the error code in the responses 530

Message Codes

Introduction

Error and information messages that are written in continuous text are much easier for human beings to read and understand, but it is considerably more difficult to have them processed by automated systems. Any change made in the texts of the messages inevitably causes extensive adjustments to the systems operated by all DENIC members. To relieve that situation, DENIC uses 11-character alphanumeric codes for the messages returned by its systems.

In this section, we describe the syntax and the structure of the 11-character alphanumeric message codes issued by the DENIC registration systems. This type of coding is applied to all newly created messages and is in consistent use in the registration system (for both .de and ENUM).

We have restricted our detailed description to the first five digits of the codes because these are the ones which are of greatest interest to the user

Contents

This chapter covers the following topics:

- Information about the Message Codes
- Structure and Components of Message Codes

Structure and Components of Message Codes

Basic Information

Every message has a single unique 11-character code to go with it. This code remains unchanged even though it is possible to vary the text of the message over time. Members are thus strongly recommended to set their systems to parse on the basis of these 11-character codes.

The codes have been structured logically and systematically, making it possible for messages to be processed in a way permitting an automatic analysis, grouping or aggregation.

General Format

Each individual position or each group of positions of the 11-character codes has its own particular meaning. The individual positions are grouped into four blocks, which together comprise the code as a whole.

0	0	0	0	0	0	0	0	0	0	0
A	B				C		D			

- A: Classification
- B: Interface
- C: Component
- D: Code

A - Classification (position 1)

The code's first position is used to classify the type of message. This is also an indication of how critical the message is.

1	INFO, success
2	INFO, failed
3	WARNING, temporary error
4	WARNING, permanent error

5	ERROR, temporary error
6	ERROR, permanent error
7	TECHNICALERROR, temporary error
8	TECHNICALERROR, permanent error

- **INFO**
  DENIC has carried out an instruction exactly as it was given, or DENIC provides information on its own initiative, not in response to an instruction.
- **WARNING**
  DENIC has executed a request, but has changed at least one detail compared with the content of the request.
- **ERROR**
  DENIC has not executed a request because there was at least one error.
- **TECHNICALERROR**
  DENIC has not been able to process the request for reasons not related to its form or content, but because of technical problems with request processing.
  - **temporary**
    The error is of a temporary nature only. You may re-submit the request later. This category includes, e.g. errors due to name servers that are temporarily inaccessible.
  - **permanent**
    The error is a fundamental one. For example: You will never succeed in registering a domain with the status "invalid", no matter when and how often you might try.

B - Interface (position 2)

Position 2 contains the code for the interface that is used.

1	currently not used
2	currently not used
3	RRI (.de)
4	RRI (.9.4.e164.arpa)
5	whois
6	not dependent on a particular interface
9	web application

B - Interface (positions 3 to 5)

The next three positions characterize the corresponding module that processed the request. Generally speaking, the names given to the modules are close to the name of the particular request form or procedure.

The expression "not dependent on a particular module" means that the message does not refer to a particular request form but has something to do with the object of the request. For example: 300 refers generally to a domain request, irrespective of whether it is a CREATE (which would be 310), an UPDATE (which would be 320) or any other request form.

000	Not dependent on a particular application
100	LOGIN / LOGOUT
200	Contact - not dependent on a particular module
210	CREATE (contact)
220	UPDATE (contact)

300	Domain - not dependent on a particular module
310	CREATE (domain)
320	UPDATE (domain)
330	CHHOLDER
340	RENEW (ENUM domain)
345	Expire (ENUM domain)
348	Expire (DE domain)
350	DELETE
360	TRANSIT
370	CHPROV
391	CREATE-AUTHINFO1
392	DELETE-AUTHINFO1
393	CREATE-AUTHINFO2
394	AuthInfo-expire
400	CHECK
450	INFO
500	RAI - not dependent on a particular module
600	QUEUE - not dependent on a particular module
610	QUEUE-READ
650	QUEUE-DELETE

Parsing the error code in the responses

Parsing for error codes

In the document's error code tables, in addition to the error codes, the associated messages and explanations are also given. Both pieces of information are human-understandable translations of the machine-readable error codes.

We recommend that the messages of the responses should not be evaluated automatically, as the error texts can be subject to constant changes. This also applies to messages that contain a dynamic error text in addition to the static error text.

Limit your automated evaluation of an error to the error code only.

Example error response in K/V format to a CREATE request

```
RESULT: success
ERROR: 53300102912 Nameserver error [ERROR: 901 Unexpected
RCODE (1)]
WARNING: 33300102900 The provided DNS data are not acceptable.
Domain "Status" is "failed". [2021-08-20T13:06:05+02:00]
STID: 7767b575-eace-11eb-9ccc-0141a1811e3e
CTID: Test#1626945804-27198095-0
```

Example error response in XML format to a CREATE request

```
<?xml version="1.0" encoding="UTF-8"?>

<registry-response xmlns="http://registry.denic.de/global/3.0"
xmlns:tr="http://registry.denic.de/transaction/3.0">
 <tr:transaction>
  <tr:stid>1ebc1fb2-eace-11eb-9ccc-0141a1811e3e</tr:stid>
  <tr:ctid>Test#1626945656-27198094-0</tr:ctid>
  <tr:result>success</tr:result>
  <tr:message level="error" code="53300102912">
   <tr:text>Nameserver error</tr:text>
   <tr:argument>ERROR: 901 Unexpected RCODE (1)</tr:argument>
  </tr:message>
  <tr:message level="warning" code="33300102900">
   <tr:text>The provided DNS data are not acceptable. Domain
"Status" is "failed".</tr:text>
   <tr:argument>2021-08-20T13:06:05+02:00</tr:argument>
  </tr:message>
 </tr:transaction>
</registry-response>
```

Sub-Title1

08 Frame Conditions for Registration

Frame Conditions for Registration 533

Frame Conditions for Registration

Introduction

In this chapter, we will explain the frame conditions you must observe when registering .de domains.

Contents

This chapter covers the following topics:

- Rules and Regulations
- Bulk Jobs
- Test Environment
- Mailing Lists
- Maintenance Work

Rules and Regulations

In this Section

In this section, we have compiled the most important rules and information for you. For the up-to-date version of the contractual rules, visit our member website.

The Rules

- Access to the Realtime Registry Interface RRI in the live environment Your access to the RRI in the live environment will not be established until you have tested all the request forms in the members' test environment.
- Faulty logins to RRI: After 10 unsuccessful attempts to log in to RRI, your login is blocked for one hour.
- Connection to RRI was terminated without logout: If you terminate the connection without logging out, the session with the RRI server will be continued for a predefined period. This may have the effect that you cannot log in for another session. One potential reason: You have already set up the maximum number of 7 concurrent sessions.

- Notes on requests
    - Unknown keywords will be ignored.
    - Wrong spelling of keyword (e.g. Deconnect instead of Disconnect of AuthInfo instead of AuthInfo).
    - The keyword and the assigned value (:) are separated by mistake by one or several blanks (Action : instead of Action:). This error occurs only with requests in the key/value format. In case of XML requests validation against the schema will be unsuccessful.
    - An UPDATE request will always be executed completely. If the update fails, the entire order will not be carried out.
    - If values are stated for an object (for instance in case of a DENIC handle) which are optional, these values will be deleted if the corresponding fields are left blank in an update (in our example of the DENIC handle).
- Notes on contacts
    - If you want to collect data of future domain holders such as name, address and e-mail address, you must inform the future domain holders of your activities in advance. You must state the purpose, scope and the intended use of the data. In addition to that you must inform your client that DENIC will also store client data when a domain is registered. You will provide this information by submitting the DENIC Domain Terms and Conditions to the client; they explicitly state this information. You may also include a corresponding note in your own terms and conditions or in your order form to draw the client's attention the client to this fact.
    - Once you have created a Contact you cannot delete or deactivate it any more. It will always persist.
    - Even if they remain partially unchanged, you must always enter values in mandatory fields when you perform an UPDATE.
    - If a Contact of a domain which bears a DISPUTE entry is established as "Holder" of the domain, you will no longer be able to change all values

(Name, Organization, Address, PostalCode, City and Country) of the domain by an UPDATE. The RegAcc will have to contact Business Services for changes.

- Domains of the "failed" status
    - If registration of a domain fails because the Nameserver Predelegation Check is not successful, the domain will be registered with the status "failed". Such a domain must be connected with the correct technical data within 30 days. If it is not connected within this period of 30 days, the domain contract will be terminated and the domain will be deleted automatically (cf. DENIC Domain Terms and Conditions § 1 (1)). Please note that you cannot re-transfer a "failed" domain to DENIC for administration.
- Important information about CHHOLDER
    - When the processing of a CHHOLDER includes the transfer of a domain to a new domain holder, all AuthInfo entries stored for the domain become invalid.
    - If a domain bears an active DISPUTE entry, holder changes (CHHOLDER) are impossible.
    - You may enter a maximum of 5 holders through the registration system.
- Important information about TRANSIT
    - Please check, if and to what extent your contractual relation with the domain holder includes operation of technical items (name server). Do not disconnect a domain you have put into TRANSIT until you have verified that the contractual relationship for providing the name services has also been terminated (Disconnect: "true"). If the name service is provided by a third party (e.g. by the client), disconnection often is not permitted or not useful.
    - When a domain is returned to DENIC, the AuthInfo1 which was stored with DENIC by the administering RegAcc becomes invalid and is deleted. The RegAcc placing the request is informed accordingly in the request acknowledgement. A possibly existing AuthInfo2 is not affected by the TRANSIT request.

- TRANSIT becomes impossible as soon as a domain has the status "failed".
- Possibly existing Dnskey-, General Request- and Abuse Contact-entries will be deleted.

- Important information about DELETE
  - "Holder" is an optional keyword in a DELETE request. If you do state a holder, the data must be identical with that recorded in the DENIC database. Otherwise the request will be rejected with an error message.
  - All other optional data (such as nserver) is ignored.
  - If the domain which is requested to be deleted bears a DISPUTE entry, it will be automatically re-registered for the DISPUTE holder immediately upon deletion. In this case the Grace Redemption Period does not apply.
  - DENIC can remove a domain from the Redemption Grace Period upon request of the domain holder. In this case the domain will be available for registration immediately.
  - Deleted Domains can either be restored with a RESTORE request or with a CHPROV with AuthInfo2.

Bulk Jobs

For bulk orders (> 5,000 orders), please contact DENIC Business Services (Kontaktinformation
Contact information) or use the members' chat to discuss the best possible procedure.

Test Environments

Overview of the test environments

- DENIC provides two separate member test environments in which all order forms can be tested.
- The test environment is an image of the live environment. Thus you can carry out tests under the same conditions as in the live environment.

Use of Test Environments

- The test environments can be accessed by every RegAcc.
- Use of the test environments is free.
- The test environments are separate environments: Contacts / domains that exist in the live environment are not taken over; you have to establish separate contacts and domains for each test environment.

RRI Server

- You need at least one separate RRI login for using RRI in the test environments.
- You need a separate RegAcc profile for the test environments.
- Test environment: https://member.denic.de/RAI-mtest/
- **Test environment**
  rri.test.denic.de (TCP-Port: 51131)

Public-whois

- The test environment is public.
- **Test environment**
  whois.test.denic.de

Provider Change in the Test Environment

Every member can have several RegAccs created in the test environments and thus test a change of provider. A separate RegAcc profile is required for each access. Send the completed RegAcc profiles with a signed e-mail to DBS (Kontaktinformation
Contact information). Please note that the IP addresses of the profiles should differ.

Testing the Effects of Domains Bearing a DISPUTE Entry

If you need domains with a DISPUTE entry for test purposes, please register the desired domains and communicate their names to DENIC Business Services by means of a key-signed e-mail.

As a matter of principle, Business Services will be the owner of all such DISPUTEs. Thus, the department will automatically become the new holder in case the domain is deleted in one of the test environments.

Mailing Lists

Information about the DENIC Mailing Lists

DENIC makes available various mailing lists, to which you may subscribe. The channels to be used for subscription are different for each list.

Public Mailing Lists

- **public-l**
  This is a public mailing list for discussing general and technical topics all around .de domains and DENIC. DENIC also publishes its press releases on this list.
- **maintenance-l**
  On this public list DENIC informs about imminent maintenance work in the live environment as well as about current disturbances.

You will find the links for subscribing to these lists on the DENIC website.

Mailing Lists for Members Only

- **DENICmembers-l**
  The DENICmembers-l is the DENIC members' official lists for discussions and for receiving information. Admittance to the list and updates of subscribers are processed by Business Services.
- **DENICtech-announce-l**
  DENIC uses the DENICtech-announce-l to inform its members about planned technical changes and innovations. You can use your RegAcc profile to subscribe to this list.

- **hostmaster-l**
  The hostmaster-l is used to discuss and exchange opinions about technical topics. You can use your RegAcc profile to subscribe to this list.
- **technews-review-l**
  On the technews-review-l, DENIC makes available an overview of the evolutions all around the topic of domains and Internet throughout the world, which is compiled by the Australian journalist David Goldstein in DENIC's name. If you want to subscribe to this list, use the corresponding link on the member website.

The list policy stipulates that only subscribers can post their opinion for discussions. In the event that a mail sender address of a person who is not a subscriber to the list shall be permitted to send mails to the list, the respective e-mail may be recorded in the DENIC database in addition to the subscribers. If you want an additional e-mail to be recorded, please address Business Services.

Maintenance

Maintenance Announcement

The maintenance announcement is made on the DENICtech-announce-l mailing list.

Regular maintenance window

- Regular maintenance of the technical services in the live and test environments takes place on Tuesdays between 4:00 p.m. and 8:00 p.m.
- The maintenance is announced at the latest on the Friday of the previous week.

Restrictions and Outages

Maintenance work that results in restrictions to the technical services will be announced on the website and via the mailing list.

Emergency Maintenance

Emergency maintenance can be carried out outside of the maintenance window and will be announced on the website and via the mailing list.

Software Rollouts

Due to DENIC's continuous delivery principle, unannounced software rollouts may occur outside of the maintenance window, which may result in short-term disconnections or a shift in the point of service between DENIC's data centers.

Sub-Title1

09 Notes on Legal Aspects

Notes on Legal Aspects 542

Notes on Legal Aspects

Introduction

In this chapter you will find information about legal aspects of domain registration.

Contents

This chapter covers the following topics:

- Changes and deletions of domains with a DISPUTE entry and
- other Legal Problems.

Changes and deletions of domains with a DISPUTE entry

Things to be observed if a domain is bearing a DISPUTE entry

§ 2 (4) of DENIC's Member Terms and Conditions stipulates:

> *"DENIC shall have the right to place a Dispute entry on a domain under .de if a third party presents a credible case to show that they have a right to the domain or that their rights have been infringed by the domain, and provided such third party declares that they intend to enforce the resulting claims against the domain holder, and provided that they also expressly declare that they will not in any way hold either DENIC or the DENIC member administering the domain liable for eventual claims by the domain holder or any third party. The Dispute entry takes effect for a year, but may be extended by DENIC if its holder is able to submit evidence that the dispute has not yet been brought to a conclusion. It is possible for the domain holder to continue to use a domain that has had a Dispute entry placed on it; it is not, however, possible to transfer it to any third party."*

This means, if a domain bears a DISPUTE entry:

- The domain can be updated by means of an UPDATE request; you may change the data of the General Request, Abuse Contact as well as the technical data entered for the domain.
- A CHHOLDER request will be rejected with an error message because you must not change the domain holder.
- A CHPROV request will be accepted, provided the mandatory keywords (Name, Organization, Address, PostalCode, City, CountryCode) remain unchanged.

- A DELETE request can be executed. The DISPUTE holder will become the new domain holder.
- A handle which is referenced as holder in a domain bearing a DISPUTE entry, cannot be updated. Please contact Business Services.
- The output in response to an INFO query concerning a Contact will include the information whether the queried contact is listed as domain holder in a domain which bears a DISPUTE entry. The value "DisputeReference" will be set to "true".
- The output in response to an INFO or whois query concerning a domain administered by the RegAcc submitting the query (own domain) will include information about possibly existing active DISPUTE entries.

Common Problems

A handle referenced as holder in several domains shall be updated. The update will fail if any - and even if it is only one - of the domains is bearing a DISPUTE entry. Since the RegAcc will be unable to find out which of its domains is concerned, it should contact Business Services.

Notice

If a domain is bearing a DISPUTE entry and the holder shall be updated (e.g. in case of a change of name or address of the company), send the corresponding evidences to DENIC. The following documents can be used as evidence:l:

- Move of a natural person: confirmation of registration at the new place of residence or similar type of document (e.g. two tax assessment notes - one addressed to the address recorded at DENIC and one addressed to the new address).
- Move of a sole proprietorship company / business: a trade report
- Move of a German GmbH / GmbH & Co. KG / AG: an extract from the commercial register

Please note that the evidences must prove both addresses and/or legal forms, the current and the future one.

Changes to Domains in the Name of DENIC's Legal Department

In certain cases, domain data will be updated by the Legal Department of DENIC. This may happen, for instance, if holder data are updated, even though a DISPUTE entry is placed on the domain.

In such cases, the RegAcc will be notified informally by e-mail. The e-mail will be sent to the address recorded for EmailLegal. If no such e-mail has been recorded, the EmailTo address will be used.

Domain Deletions in the Name of DENIC's Legal Department

In certain cases, a domain will be deleted at the instigation of DENIC's Legal Department. This may happen, for instance if the domain infringes the Domain Terms and Conditions or a judgement on the substance of the case has been issued.

In this case, the RegAcc is notified by a message in its message queue (if MsgFeed “rri” is used) or by an e-mail to the address recorded for EmailTo (if MsgFeed “mail” is used).

> **Notice**
>
> Please note that domain deletions instigated upon request of DENIC’s Legal Department (DENICrecht) will be executed without delay. There is no Redemption Grace Period.

Other Legal Problems

Infringements of the Domain Terms and Conditions

The RegAcc should always make sure that the domain data is complete and correct. This is in the interest of the RegAcc itself and of the domain holder. Particularly important is that the domain holder is denominated clearly and uniquely. A mistake as minor as a typing error in the mandatory data may have the effect that the domain holder loses their domain.

Trotzdem kommt es immer wieder vor, dass DENIC von Dritten auf unzureichende Daten zu einer Domain aufmerksam gemacht wird. Diese unzureichenden Daten stellen einen Verstoß gegen die DENIC-Domainrichtlinien dar.

In this case:

- Business Services first informs the administering RegAcc via the e-mail address specified for “EmailLegal” (if one has been provided, otherwise via the e-mail address for “EmailTo”) and sets a deadline for the data to be corrected.
- If the data is not corrected by the deadline, the procedure is forwarded to DENIC's legal department.
- If necessary, the domain is then terminated due to a violation of the domain guidelines and subsequently deleted.

Sub-Title1

10 Annexes

Annexes 547
.de Registry Lock 547
GnuPG and PGP 560
DNSSEC 578
Network Byte Order 582
Reading and Sending Data 583
List of Message Codes 594
contact CREATE and contact UPDATE 594
domain CREATE 595
domain CHHOLDER 596
domain CHPROV 597
domain RESTORE 598
List of Message Codes for Verification 616
Link Collection 630

Annexes

Introduction

The Annexes will provide you with additional information.

Contents

The Annexes cover the following topics:

- ".de Registry Lock" below
- "GnuPG and PGP" on page 560
- "DNSSEC" on page 578
- "Network Byte Order" on page 582
- "Reading and Sending Data" on page 583
- List of Message Codes
- List of Message Codes for Verification
- "Link Collection" on page 630

.de Registry Lock

What is a Lock?

The lock (DENIC product name ".de Registry Lock") for a domain prevents manipulative, unwanted changes to the domain data.

.de Registry Lock is a paid product that members offer domain owners.

Involved

- **Domain Owner**
  The domain owner specifies exactly one lock contact for a domain for which a lock is set up.
- **Lock Contact**
  The lock contact must be a natural person and approves requests for a locked domain.

- **Mitglied**
  For a locked domain, the member remains the contact person for the domain owner and is involved in the changed processes for a locked domain.
- **DENIC**
  DENIC manually checks all orders (forms and e-mail) and carries out the changes. Finally, the participants are informed about the result. DENIC can also have the authenticity of a request for a locked domain confirmed by the lock contact.

Particularities

- Contacts that are used for a locked domain are protected against changes by contact requests to the RRI.
- At .de Registry Lock, requests that change the domain data are sent via email to DENIC and INFO requests are sent to the RRI server. Requests that change the domain data are rejected by the RRI server with an error message.

Request Forms mentioned in the Descriptions for easier Orientation

In this chapter, we give the underlying requests, in line with the locked domain workflows, for easier orientation, for example in the headings such as "CHHOLDER and UPDATE with a locked domain".

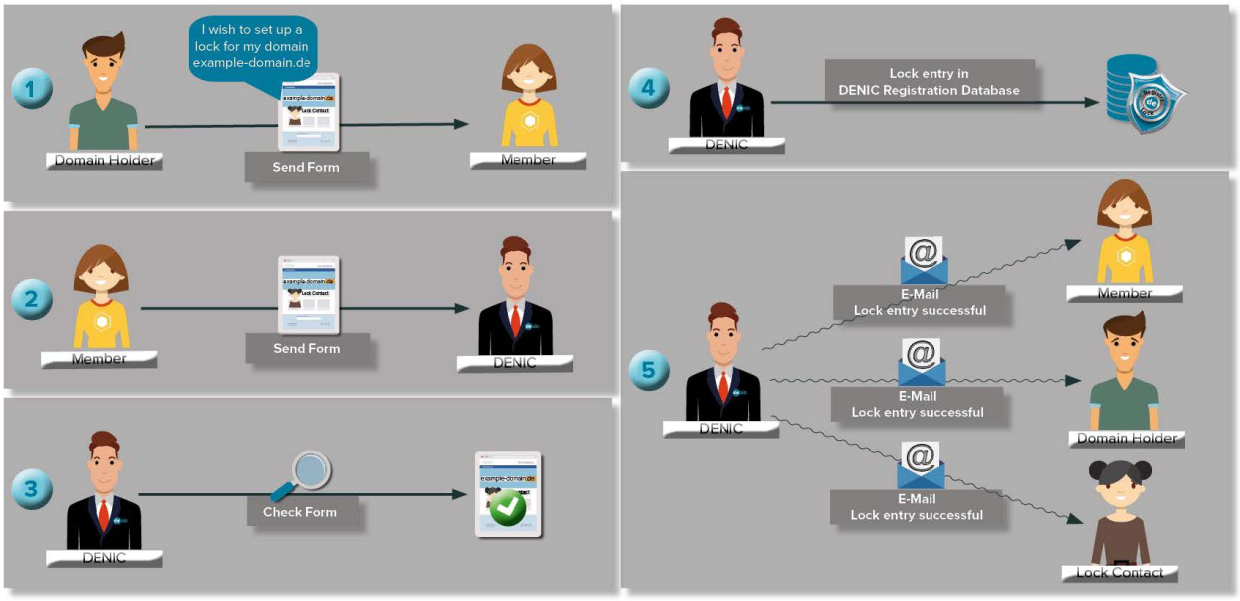
Which Requests are affected by a changed Procedure for a locked Domain?

- Domain CHHOLDER
- Domain UPDATE
- Domain CHPROV
- Domain TRANSIT
- Domain DELETE
- Contact UPDATE

Set up Lock

If a lock is set up (Lock CREATE), the domain owner (or his representative), you as a member, Lock Contact and DENIC are involved. As soon as the lock has been set up, the domain is protected against changes via the RRI.

Schematic overview - Set up Lock



Step-by-Step - Set up Lock

1. The domain owner fills out the set up lock form and sends it to you. You can find the form at:
    - German and English
    - In addition to the domain data, the name, the mobile phone number and the e-mail address of the lock contact are recorded.
    - The lock contact confirms its details with an attached proof of identity.
    - If the lock for the domain is requested by a representative of the domain owner, proof of the representation must be provided.
2. Please check that the form is correct and that all necessary evidence has been attached before you send the form to DENIC.

> **Notice**
>
> Send the form as an e-mail attachment to DBS (Kontaktinformation
>
> Contact information) and sign the e-mail with your master key.

3. DENIC will check the information in the form and the evidence and will contact you, if corrections are necessary or the evidence is insufficient.
4. DENIC then enters the lock in the registration database.
5. DENIC informs you, the domain owner and the Lock Contact about the successful establishment of the lock.

Remove Lock

When a domain owner removes the lock for a domain (Lock DELETE), the workflow for the domain shifts back to RRI after the removal and the lock contact is no longer responsible for the domain.

Step-by-step - Remove Lock

1. The domain owner fills out the form to remove the Lock and sends it to you. You can find the form at:
    - German and English
    - The domain data is required in the form.
    - If the request is made by a representative of the domain owner, proof of the representation must be provided.
2. Please check that the form is correct and that all necessary evidence has been attached before you send the form to DENIC.

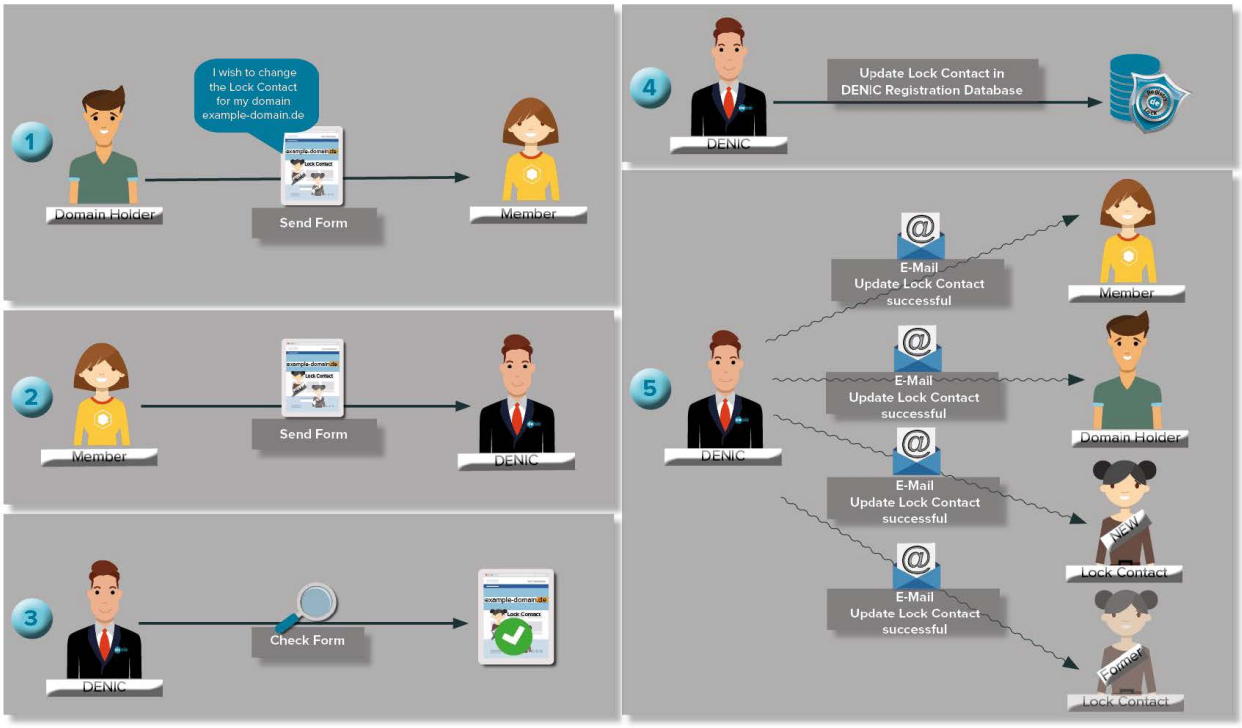
> **Notice**
>
> Send the form as an e-mail attachment to DBS (Kontaktinformation
>
> Contact information) and sign the e-mail with your master key.

3. DENIC will check the information in the form and the evidence and will contact you, if corrections are necessary or the evidence is insufficient.
4. DENIC informs you, the domain owner and the lock contact about the removal of the lock.

Update Lock Contact

The lock contact is named by the domain owner, when setting up the lock and confirms the authenticity of a request for a locked domain. The domain owner can designate someone else as lock contact at any time. The lock for a domain remains unchanged, when changing a lock contact.

Schematic overview - Update Lock Contact



Step-by-Step - Update Lock Contact

1. The domain owner fills out the Lock Contact update form and sends it to you. You can find the form at:
    - German and English
    - In addition to the domain data, the name, the mobile phone number and the e-mail address of the new lock contact are recorded.
    - The new lock contact confirms its details with an attached proof of identity.
    - If the update of the lock contact is requested by a representative of the domain owner, proof of the representation must be provided.
2. Please check, that the form is correct and that all necessary evidence has been attached before you send the form to DENIC.

    > **Notice**
    >
    > Send the form as an e-mail attachment to DBS (Kontaktinformation
    > Contact information) and sign the e-mail with your master key.

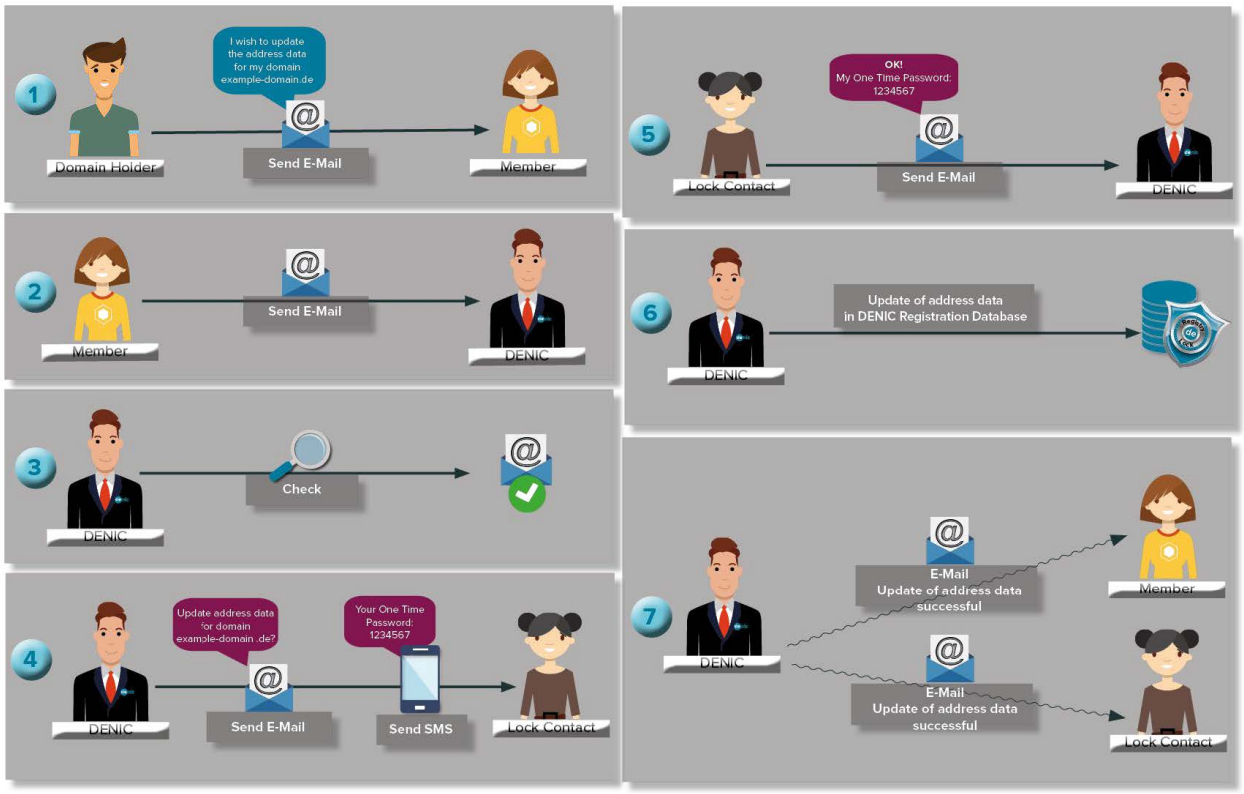
3. DENIC will check the information in the form and the evidence and will contact you, if corrections are necessary or the evidence is insufficient.
4. DENIC then enters the new lock contact in the registration database
5. DENIC informs you, the domain owner and the old and new lock contact about the change of the lock contact data.

Update Domain Data (CHHOLDER/UPDATE) with a locked Domain

- The domain data for a locked domain can be changed to the same extent as for CHHOLDER or UPDATE requests.
- The lock contact confirms the changes in the request of the domain owner, so that DENIC can make the changes.

- Incorrect name server data or non-functioning name servers have no influence on the lock and do not release the lock.

Schematic overview - Update Domain Data (CHHOLDER/UPDATE) with a locked Domain



Step-by-step - Update Domain Data (CHHOLDER/UPDATE) with a locked Domain

1. The domain owner sends you a request for changing the domain data (e.g. by e-mail).
2. You can check the request in advance and then send it to DENIC in an informally format or a key/value format in an e-mail.

> **Notice**
>
> Please do not send DENIC a request for a locked domain as an XML document. Sign the e-mail with your master key.

3. DENIC will check the information in the e-mail and will contact you if corrections are necessary.

> **Notice**
>
> Please, also send corrections in an informally format or in a key/-value format and sign your e-mail with the master key. Please, do not send any XML document.

4. The lock contact receives two messages from DENIC:
    - an e-mail with the domain data to be changed and the request to confirm the change by specifying a one-time password (OTP) and
    - an SMS with the OTP.
5. The lock contact has 7 working days (Monday to Friday with the exception of 24.12., 31.12. and public holidays in Hesse, Germany) to inform DENIC of his OTP for a cross-check and to confirm or reject the request, via (Kontaktinformation Contact information). If the Lock Contact does not respond within 7 working days (Monday to Friday with the exception of 24.12., 31.12. and public holidays in Hesse, Germany), the request expires.
6. If the order has been successfully completed, DENIC's domain data will be updated in the registration database.
7. DENIC informs you and lock contact via e-mail about the successful updates of the domain data.

> **Notice**
>
> If necessary, you can inform the domain owner about the completion of the order. The exact time of the change can be traced in the "Changed:" field via a Domain INFO request.

CHPROV with a locked Domain

The lock is removed before a provider change, since the necessary changes to the domain data (e.g. new handles or name servers) would otherwise not be possible. If the lock for the domain is to be continued, the lock must be set up again with the member who takes over the domain in its inventory and offers the .de registry lock.

TRANSIT with a locked Domain

If you want to give up the administration of a domain with .de Registry Lock, this is only possible directly via DENIC. Domains with .de registry lock can only be entered into TRANSIT manually via DENIC. In addition, disconnection is not possible in this case. DENIC will remove the lock and inform the domain owner as well as the lock contact.

DELETE with a locked Domain

Deletion of a domain must be confirmed by the lock contact. The lock will be removed by deleting the domain. Alternatively, the lock can be removed first and then the domain can be deleted.

The RGP for the domain then takes place.

Domain INFO with a locked Domain

Abfrage und Antworten bei einer Locked Domain

Queries and answers for a locked domain continue to be sent to and received from the RRI.

> **Notice**
>
> There is no "RegistryLock: false" entry for all domains without a lock. A value for the field is only output if a lock is set. Query for a locked domain in Key/Value Format.

Request for a locked Domain in Key/Value Format

```
Version: 3.0
Recursive: true
AuthInfo <Passwort>
Domain: denic.de
```

Response for a locked Domain in Key/Value Format

```
RESULT: success
STID: 4e5bfed2-80b1-11ea-b1ca-639027db68ba

Domain: de-registrylock.de
Domain-Ace: de-registrylock.de
Nserver: ns1.denic.de.
Nserver: ns2.denic.de.
Nserver: ns3.denic.de.
Status: connect
RegistryLock: true
RegAccId: DENIC-1000006
RegAccName: DENIC eG
Changed: 2020-04-23T09:58:11+02:00

[General Request]
Type: REQUEST
URI-Template: mailto:example@denic.de
Changed: 2018-08-13T14:14:18+02:00

[Abuse Contact]
Type: REQUEST
URI-Template: mailto:abuse@denic.de?subject=domain:{Ulabel}
Changed: 2018-08-13T14:14:18+02:00

[Holder]
Type: ORG
Name: DENIC eG
Address: Theodor-Stern-Kai 1
City: Frankfurt am Main
PostalCode: 60596
CountryCode: DE
Email: info@denic.de
Changed: 2017-01-06T15:51:08+02:00
```

Request for a locked Domain as an XML Document

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<registry-request xmlns="http://registry.denic.de/global/3.0"
xmlns:domain="http://registry.denic.de/domain/3.0">
 <domain:info recursive="true">
  <domain:handle>denic.de</domain:handle>
  <domain:authInfo>Passwort</domain:authInfo>
 </domain:info>
</registry-request>
```

Response for a locked Domain as an XML Document

```
<?xml version="1.0" encoding="UTF-8"?>

 <registry-response xmlns="http://registry.denic.de/global/3.0"
xmlns:tr="http://registry.denic.de/transaction/3.0">
  <tr:transaction>
   <tr:stid>d5c0f12a-aac9-11e8-afa4-f7fda02f01db</tr:stid>
   <tr:result>success</tr:result>
    <tr:data>
     <domain:infoData xmlns:-
domain="http://registry.denic.de/domain/3.0">
      <domain:handle> de-registrylock.de </domain:handle>
      <domain:ace> de-registrylock.de </domain:ace>
      <domain:status>connect</domain:status>
      <domain:registryLock>true</domain:registryLock>
      <domain:regAccId>DENIC-1000006</domain:regAccId>
      <domain:regAccName>DENIC eG</domain:regAccName>
      <domain:contact role="generalrequest">
       <contact:handle xmlns:-
contact="http://registry.denic.de/contact/3.0">DENIC-1000006-
GENERAL-REQUEST</contact:handle>
       <contact:type xmlns:-
con-
tact="http://registry.denic.de/contact/3.0">REQUEST</contact:type>
       <contact:uri-template
xmlns:-
con-
tact-
="h-
ttp://re-
gistry.denic.de/contact/3.0">mailto:example@denic.de</contact:uri-
template>
       <contact:changed xmlns:-
contact="http://registry.denic.de/contact/3.0">2018-08-
13T14:14:18+02:00</contact:changed>
      </domain:contact>
      <domain:contact role="abusecontact">
       <contact:handle xmlns:-
contact="http://registry.denic.de/contact/3.0">DENIC-1000006-ABUSE-
CONTACT</contact:handle>
       <contact:type xmlns:-
con-
```

```
tact="http://registry.denic.de/contact/3.0">REQUEST</contact:type>
       <contact:uri-template xmlns:-
con-
tact-
="h-
ttp://re-
gistry.denic.de/contact/3.0">mailto:abuse@denic.de?subject=domain:
{Ulabel}</contact:uri-template>
       <contact:changed xmlns:-
contact="http://registry.denic.de/contact/3.0">2018-08-
13T14:14:18+02:00</contact:changed>
      </domain:contact>
     <domain:contact role="holder">
      <contact:handle xmlns:-
contact="http://registry.denic.de/contact/3.0">DENIC-1000006-
DENIC</contact:handle>
      <contact:type xmlns:-
contact="http://registry.denic.de/contact/3.0">ORG</contact:type>
      <contact:name
xmlns:contact="http://registry.denic.de/contact/3.0">DENIC eG</-
contact:name>
      <contact:postal
xmlns:contact="http://registry.denic.de/contact/3.0">
       <contact:address>Theodor-Stern-Kai 1</contact:address>
       <contact:postalCode>60596</contact:postalCode>
       <contact:city>Frankfurt am Main</contact:city>
       <contact:countryCode>DE</contact:countryCode>
      </contact:postal>
      <contact:email xmlns:-
con-
tact-
="h-
ttp://registry.denic.de/contact/3.0">info@denic.de</contact:email>
      <contact:changed
xmlns:contact="http://registry.denic.de/contact/3.0">2017-01-
06T15:51:08+02:00</contact:changed>
     </domain:contact>
     <dnsentry:dnsentry xmlns:dnsentry-
y="http://registry.denic.de/dnsentry/3.0" xmlns:x-
si="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="dnsentry:NS">
      <dnsentry:owner>de-registrylock.de.</dnsentry:owner>
      <dnsentry:rdata>
       <dnsentry:nameserver>ns1.denic.de.</dnsentry:nameserver>
      </dnsentry:rdata>
```

```
nsaction>
</registry-response>
```

Error Message with a locked Domain

Locked Domain - Error Table

If a lock has been set for a domain, all requests to the RRI that change the domain data are blocked by the same error message:

Error Code	Error Mes-sage	Input Format	Update / Chholder / Delete / Chprov / Transit	Brief Description
53000080009	Request rejected - this domain is locked!	KV / XML	all	The request cannot be carried out via the RRI because a lock was set on the domain..

GnuPG and PGP

Basic Information about GnuPG and PGP

Introduction

Each RegAcc must record at least one key with DENIC.

What is GnuPG / PGP?

PGP (Pretty Good Privacy) / GnuPG (an Open Source / Open PGP Implementation) are programs which make it possible to sign and encrypt e-mails, texts and files. It is a technique considered to offer a very high level of security and thus to permit secure communication between two parties.

Supported Versions

DENIC supports all versions of GnuPG and PGP.

What Is Meant by Keys and how Are they Structured?

A key is always comprised of two parts:

- the public key and
- the private key.

As you will realize, this encryption procedure is an asymmetrical one. So each user has two keys which belong together, the private one and the public one. GnuPG/PGP generates a so-called hash value (secret key) out of the selected password. The hash value or secret key, in turn, is used to generate the private part of the key.

Two of the things you can do with your private key are to sign messages and decrypt them.

The public key can be used to verify messages and encrypt them.

Each key contains various items of information, such as:

Length	The length of the key, for example 2048 bits
Type	For example: DSA and ElGamal
KeyID	The key's unique identifier, for example: 0x00F555A
Date of creation	Dates are written: yyyy-mm-dd
Name	For example: DENIC-1000002 Communication Key for Production
e-mail	For example: example@denic.de
Fingerprint	The key's "fingerprint", which permits unique identification in combination with the KeyID
Expire	The date on which the key will expire, if stated

Let us take an example key:

```
sheridan@babylon5-xs2:~/ $ gpg --fingerprint F81AE61F
pub 2048D/F81AE61F 2001-12-11 [verfällt: 2002-12-11]
   Schl.-Fingerabdruck = 4943 3AA7 6A85 306E 23A4 A1AC D4B9 6CF6
F81A E61F
uid          Business Services <example@denic.de>
uid          DENICoperations <ops@denic.de>
uid          Public Relations <presse@denic.de> sub
2048g/839B30DB 2001-12-11 [verfällt: 2002-12-11]
```

In this example:

- Length: 2048 Bit
- Type: DSA (symbolized by the "D" in 2048D; if RSA keys were used, it would be for example 2048R)
- KeyID: 0xF81AE61F
- Created on: 2001-12-11 (ISO format for dates: YYYY-MM-DD)
- Name: Business Services
- E-mail: example@denic.de
- Fingerprint: 4943 3AA7 6A85 306E 23A4 A1AC D4B9 6CF6 F81A E61F
- Expire: The key expires 2002-12-11

GnuPG / PGP save keys in a file or set of files known as a "keyring".

Usually, a keyring is comprised of two files: "pubring.pkr" and "secring.pkr" (for PGP) or "pubring.gpg" and "secring.gpg" in $HOME/.gnupg/ (for GnuPG).

Master and Communication Keys

DENIC differentiates between two types of keys:

- master key
- communication key

The master key is needed to

- sign e-mails to request access to the Registrar Administration Interface (RAI) at DENIC Business Services
- authorize changes to the stored RAI info e-mail address to which DENIC sends notifications about changes to the RegAcc profile
- authorize additional orders of RegAccs/contingents
- authorize the creation of accesses to the member websites

The master key or the communication key is needed to

- communicate between members and DENIC via e-mail on general topics

Function and requirements of the master key and communication key

The key is used for authorization and identification vis-à-vis DENIC when processing orders and other e-mail communication. Each RegAcc must store at least one key.

- Each key must be unique. A key that has already been used cannot be reused.
- The number of keys allowed is not limited.
- All keys must have a validity period of no more than 731 days. After the validity period has expired, the key is deleted from DENIC's keyring and can no longer be used.
- DH/DSS, RSA, DSA or DSA and Elgamal must be selected as the key type.
- The key length must be at least 2048 bits.

- The signature algorithms permitted for requests are:
    - RIPEMD160withRSA
    - RIPEMD160withECDSA
    - RIPEMD256withRSA
    - SHA256withDSA
    - SHA256withECDSA
    - SHA384withDSA
    - SHA384withECDSA
    - SHA512withDSA
    - SHA512withECDSA
    - SHA256withRSA
    - SHA384withRSA
    - SHA512withRSA
- Each key should be labeled with a secure passphrase.
- DENIC recommends choosing a unique name for each key, consisting of RegAccId, RegAccName and, if applicable, the purpose of use.
    - Example: DENIC-99995 DENIC eG.

Activating a Key

A new key can be managed via RAI. The key will be active within 5 minutes after being inserted.

Expire of a Key

Expired keys will be removed from the DENIC keyring and will no longer be displayed in RAI.

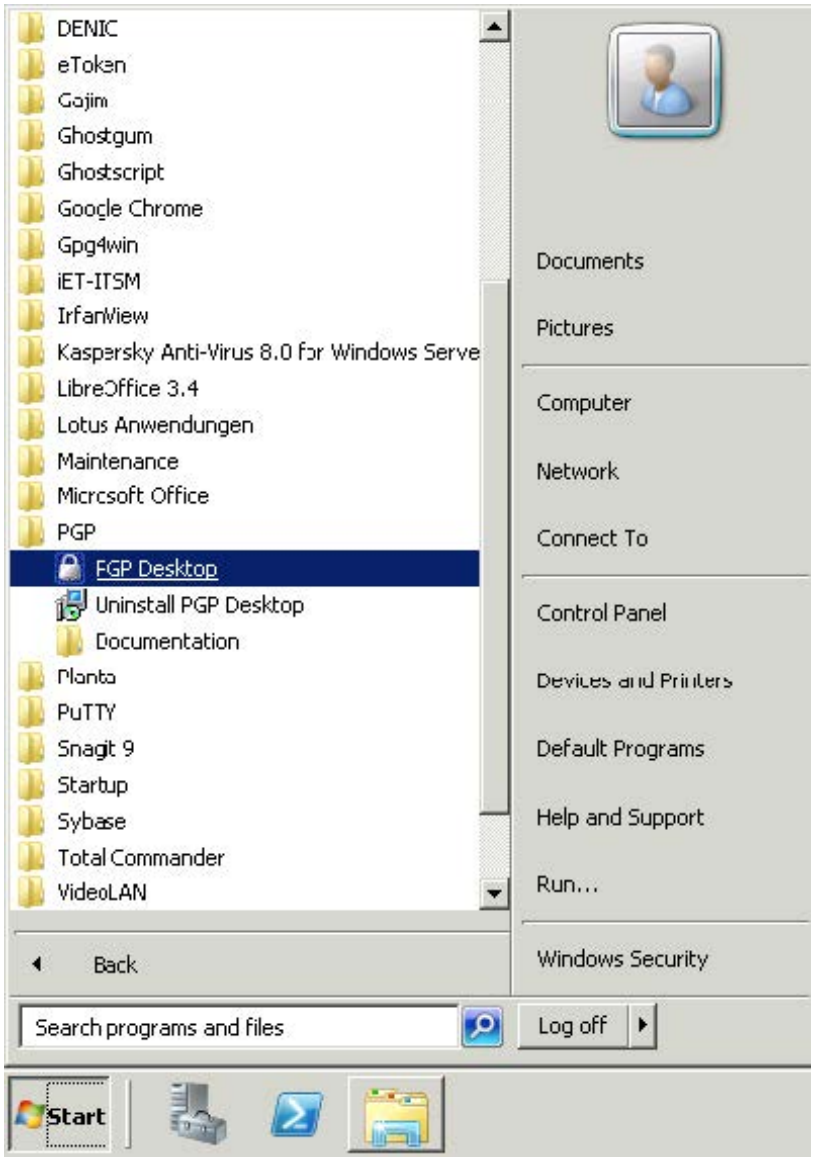
Deleting a Key

If you wish to remove a key from the DENIC keying (e.g. because the key has been compromised), you can remove it from the DENIC keying at any time using the RAI.

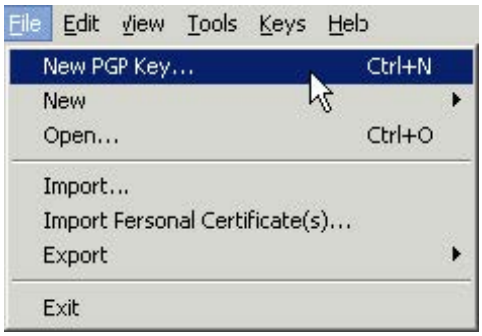
Step by Step: Creating a Key with PGP

These instructions are based on the PGP version 9.9.1.

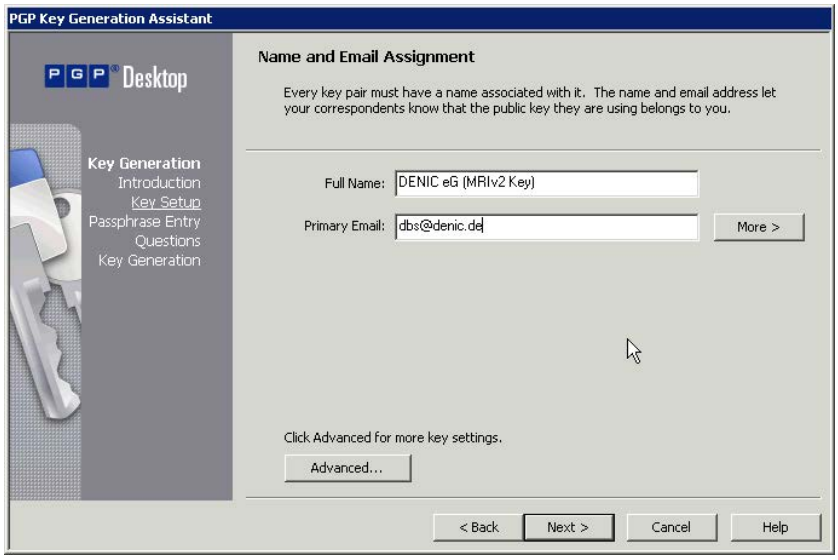
- Start the PGP-Desktop.



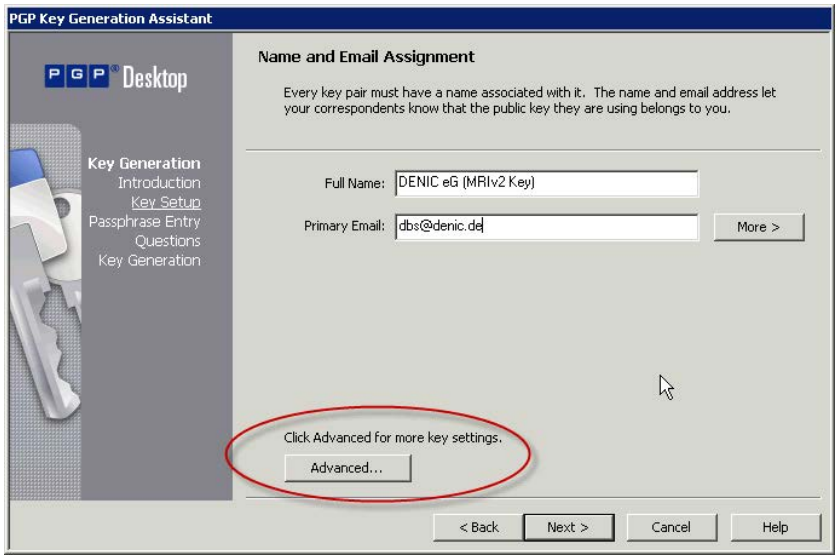
- Select New PGP Key in the File menu



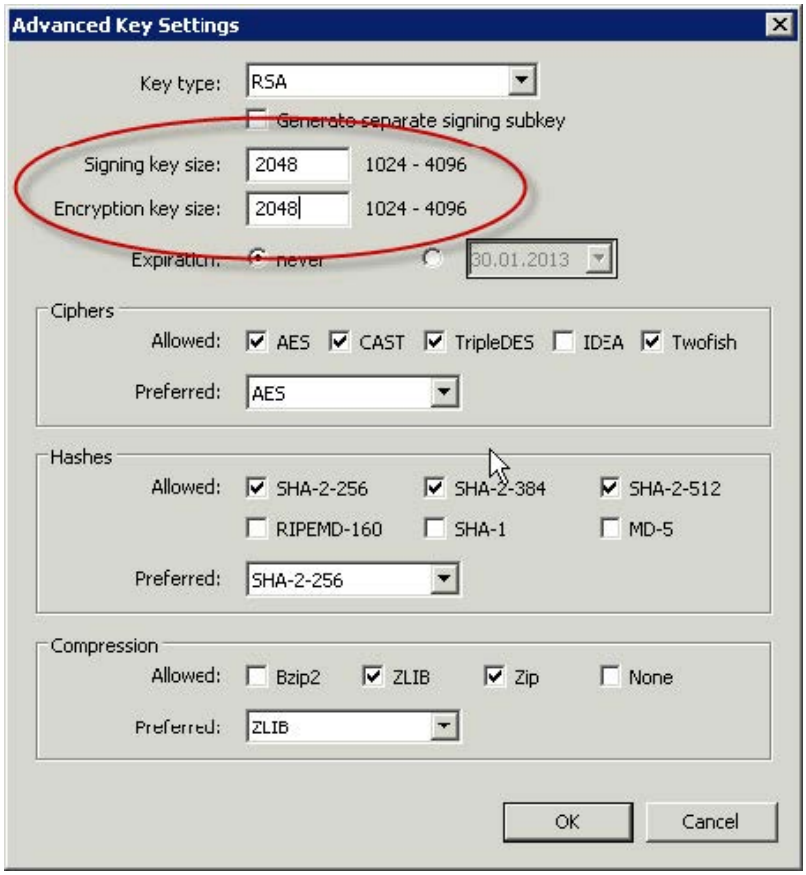
- Enter the name of the person / role who is going to use the key and the associated e-mail address



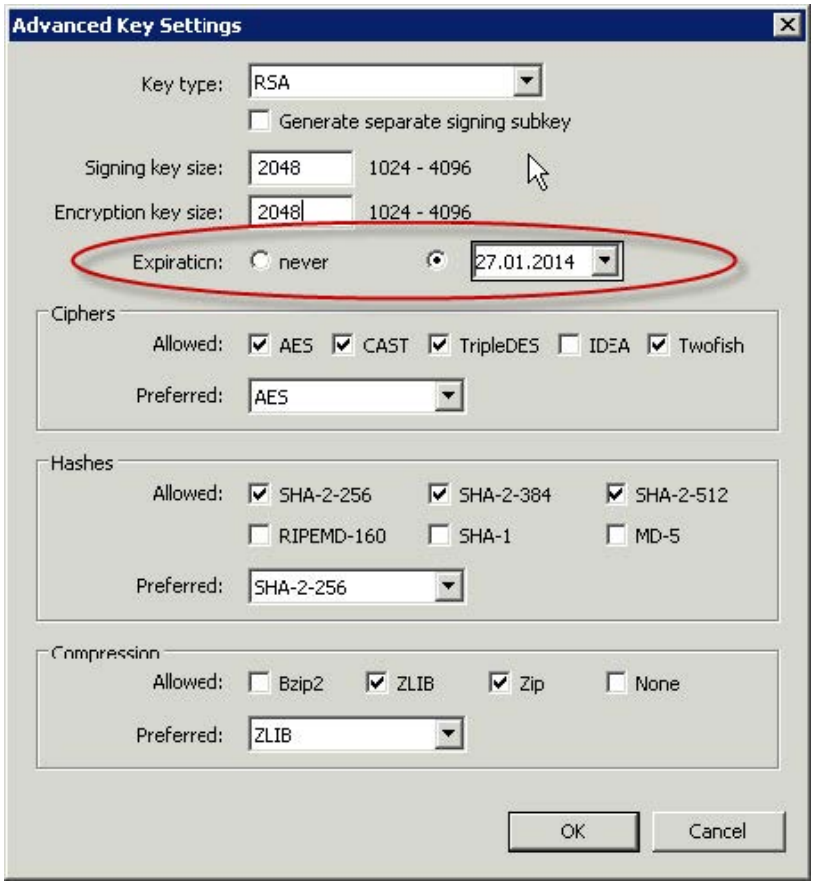
- Click "Advanced"



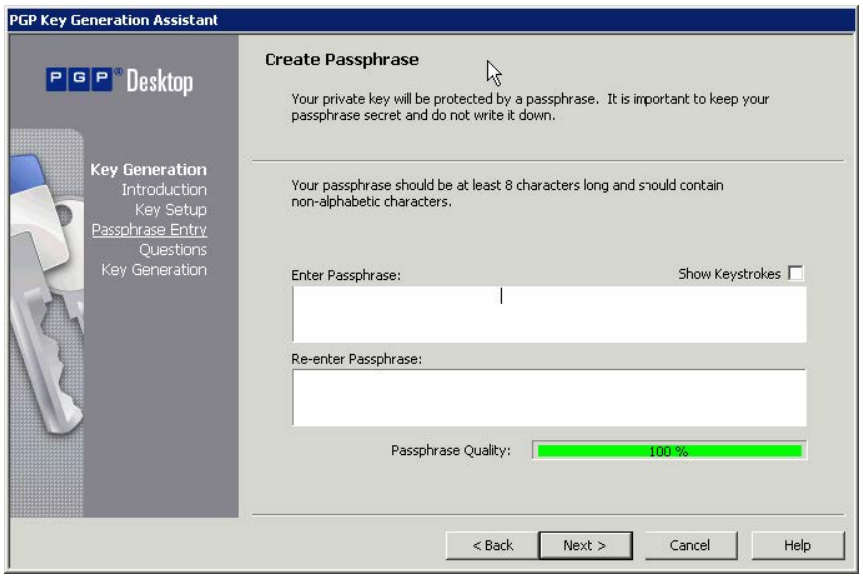
- Check whether the value displayed for the key size is at least "2048"



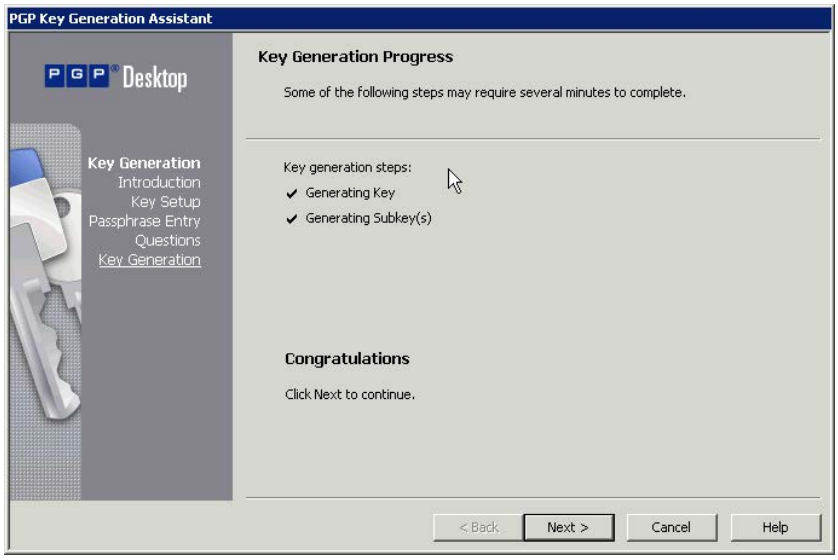
- For the expire date click the date field and enter a date that is no more than 731 days in the future



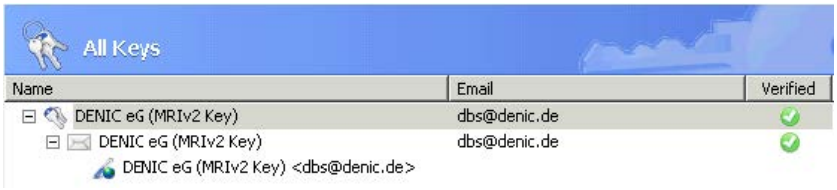
- Click "OK" and "Next" and enter a secure passphrase



- Click "Next" and your pair of keys is generated

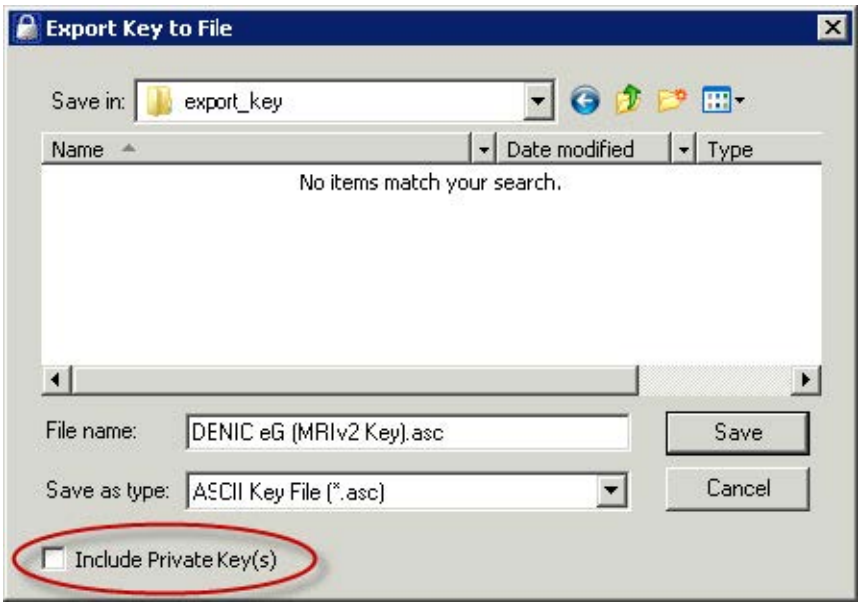


- After that, you have the option of uploading your key to a public keyserver if you wish.
    - Then right-click on your key



- Wählen Sie "Exportieren" aus.

- Select where to store the key file and make sure that you have NOT selected "Include Private Key(s)"!



- Save the created key in a save location.

Notice

Please note that the key file also contains the private key. Store the private key carefully as it will be used for signing.

Incorporating the Communication Key via the Registrar Administration Interface

- New communication keys can be managed via RAI.
    - Detailed information on this topic can be found in the documentation "RAI - Registrar Administration Interface", How to Edit the RegAcc Profile, "PGPKey".
- Open the key file with a text editor.
- Select and copy only the public part of the key.

- Insert the text block in RAI under PGPKey and save your action.

> **Notice**
>
> Now you can use the key within 5 minutes.

Step by Step - Creating a Key with GnuPG

Creating a Key

These instructions are based on GnuPG, version 1.4.7. Follow the procedure described below to create a key:

Linux

- Start a computer console (by pressing ALT+F2 and enter "konsole" under KDE and "gnome-terminal" under GNOME).

Windows

- Open a command window (Start -> Run -> cmd)
- Type in "gpg --gen-key"

```
[chang@theorchid]~ gpg --gen-key
gpg (GnuPG) 1.4.10; Copyright (C) 2008 Free Software Foundation, Inc.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection?
```

- Now select what kind of key you want (DENIC recommends to select either "RSA and RSA (default)" or "DSA and Elgamal").

- Now enter the key size. Please enter at least 2048 but no more than 4096 bit.

```
Please select what kind of key you want:
   (1) RSA and RSA (default)
   (2) DSA and Elgamal
   (3) DSA (sign only)
   (4) RSA (sign only)
Your selection? 1
RSA keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048) 4096
```

- Specify how long the key should be valid:
    - You may state the term of validity for the key in days (Enter, for example, 10 for 10 days) or
    - in weeks (Enter, for example 4w for 4 weeks) or
    - in months (Enter, for example, 12m for 12 months) or
    - in years (Enter, for example, 2y for 2 years)

- > **Notice**
  >
  > DENIC recommends to enter the maximum validity of 2years.

- Bestätigen Sie im Anschluss Ihre Eingabe.

```
Please specify how long the key should be valid.
         0 = key does not expire
       <n>  = key expires in n days
       <n>w = key expires in n weeks
       <n>m = key expires in n months
       <n>y = key expires in n years
Key is valid for? (0) 2y
Key expires at Sun Jan 26 11:19:43 2014 CET
Is this correct? (y/N) y
```

- Enter the name of the person / role who is going to use the key and the associated e-mail address

```
You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: DENIC eG (MRI v2 Key)
Email address: dbs@denic.de
Comment:
You selected this USER-ID:
    "DENIC eG (MRI v2 Key) <dbs@denic.de>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?
```

- Confirm your entry again

```
You selected this USER-ID:
    "DENIC eG (MRI v2 Key) <dbs@denic.de>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.
```

- Enter a secure passphrase. Now your key is generated

```
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
.....................................+++++
..+++++
We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
....................+++++
+++++
gpg: key D8C70FC1 marked as ultimately trusted
public and secret key created and signed.

gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   5  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 5u
gpg: next trustdb check due at 2014-01-26
pub   4096R/D8C70FC1 2012-01-27 [expires: 2014-01-26]
      Key fingerprint = D296 45FA 113A 18A3 85E0  EC55 F9E7 22EF D8C7 0FC1
uid                  DENIC eG (MRI v2 Key) <dbs@denic.de>
sub   4096R/66D08FB1 2012-01-27 [expires: 2014-01-26]
```

- You may now export your key by typing in gpg -a --export <KeyID> > mykey.asc. You will fin the key ID in the pub line at the top

```
pub   4096R/1C040193 2011-12-19 [verfällt: 2013-12-18]
      Schl.-Fingerabdruck = 2EEE 2DE1 6F3E 4442 3D6D  0ACC E108 2B08 1C04 0193
uid                  DENIC eG (MRIv2 Key) <dbs@denic.de>
sub   4096R/F3BD07B9 2011-12-19 [verfällt: 2013-12-18]
```

- Please also export the private key with gpg-a – export-secret-keys<KeyID> >mysecretkey.asc.
- To conclude the action, please save your pair of keys in a save location.

> **Notice**
>
> Please note that the key file contains the private key. Be careful to store the private part in a secure place, since it is used for signing.

Incorporating the Communication Key via the Registrar Administration Interface

- New communication keys can be managed via RAI.
    - Detailed information on this topic can be found in the documentation "RAI - Registrar Administration Interface", How to Edit the RegAcc Profile, "PGPKey".
- Open the key file with a text editor.
- Select and copy only the public part of the key.

- Insert the text block in RAI under PGPKey and save your action.

> **Notice**
>
> Now you can use the key within 5 minutes.

Further Information About GnuPG / PGP

List of important GnuPG commands

-s, --sign	Create signature
--clearsign	Create cleartext signature
-e, --encrypt	Encrypt file / e-mail

--decrypt [file]	Decrypt file
--verify [signature] [file]	Verify signature
--list-keys	List all keys
--list-public-keys	List all public keys
--list-secrect-keys	List all private keys
--list-sigs	List all keys with their signatures
--check-sigs	Like --list-sigs, but including verification of signature
--fingerprint	Like --list-keys, but including output of fingerprint
--gen-key	Delete key from keyring
--delete-key	Schlüssel aus dem Keyring entfernen
--delete-secret-key	Delete public and private key from keyring
--export [KeyID]	Export public key
--export-secret-keys [KeyID]	Export private key
--import	Import public key into keyring
-a, --armor	Output in"ASCII-armor"
-o, --output [file]	Transfer output to file
-o [file] –a –export [KeyID]	Output public key in form of ASCII-armor in file

RFCs relating to GnuPG / PGP

RFC1321	The MD5 Message-Digest Algorithm
RFC2104	HMAC: Keyed-Hashing for Message Authentication
RFC3156	MIME Security with OpenPGP

RFC4880	OpenPGP Message Format

DNSSEC

Motivation

DNSSEC aims at closing security holes in the Internet such as cache poisoning and DNS spoofing.

For validating DNSSEC a chain of trust must be established. To meet this requirement, the DNSSEC protocol requires that a reference to the key(s) of the delegated zone is entered into the delegating zone. Thus, the chain of trust follows the delegation path. The relevant key is the Key Signing Key of the delegated zone, which is usually flagged as Secure Entry Point (SEP). This information is available in a DNSKEY-RR in the delegated zone and is signed there (at least) by this very DNSKEY. It is not repeated in exactly the same way in the delegating zone for reasons of space. Instead of the actual key, a corresponding fingerprint is stored there in a DS-RR (Delegation Signer).

The corresponding DS records are generated together with the zone (currently precisely one DS-RR per communicated Trust Anchor), signed and distributed with the zone.

Dnskey

DNSKEY Resource Records serve for propagating public keys throughout the DNS and represent a new technical attribute of a domain (comparable to “Nserver” or “Nsentry” attributes).

Structure of the DNSkeys in K/V and XML requests

The following four fields are required as values of the keyword "Dnskey:" (cf. presentation format in RFC 4034):

- Flags: 256 or 257,
- Protocol: currently always 3,

- Numerical encoding of the used algorithm:
  - The values in the “Algorithm” field must be taken from the IANA-Registry, marked as usable and not reserved for private purposes. Only RSA, DSA, ECDSA and GOST with different hash procedures are defined as cryptographic algorithms:

3	(DSA/SHA1)
5	(RSA/SHA-1)
6	(DSA-NSEC3-SHA1)
7	(RSASHA1-NSEC3-SHA1)
8	(RSA/SHA-256)
10	(RSA/SHA-512)
12	(GOST)
13	ECDSAP256SHA256)
14	(ECDSAP384SHA384)

- Public-Key, base64-kodiert.

The value of the “Dnskey” keyword must be entered on one line and must not be separated by line breaks.

Domain CREATE request example with specification of a Dnskey in K/V format

```
Action:    create
Version:    3.0
CtId:       cba-987654321
Domain:     beispiel-fünf.de
Domain-ace: xn--beispiel-fnf-mlb.de
Holder:     DENIC-99990-BSP
Nserver:    ns1.provider.de
Nserver:    ns2.xn--tste-loa.de 192.0.2.1
Dnskey:     257 3 5 AwEAAdDECa-
jHaTjfSoNTY58WcBah1BxPKVIHBz4IfLjfqMvium4lgKtKZLe97DgJ5/NQrNEGGQ
r6fKvUj67c-
frZUo-
jZ2cGR-
izVhgkOqZ9s-
caTVXNuXLM5Tw7VWOVIceeXAuuH2mPIiEV6MhJYUsW6d-
vmNsJ4XwCgNgroAmXhoMEoWEj
BB+wjYZQ5GtZHBFKVXACSWTiCt-
ddHcueOeSVPi5WH94Vlub-
hHfiytNPZLrOb-
hUCHT6k0tNE6phLoHnXWU+6vpsYpz6GhMw/R9BFxW5PdPFIWBgoWk2/XFVRSKG9Lr61-
b2z1R126xeUwvw46RVy3hanV3vNO7LM5HniqaYclBbhk=
```

Domain CREATE order example with specification of a DNS key in XML format

```
<?xml version='1.0' encoding='UTF-8'?>

<registry-request xmlns='http://registry.denic.de/global/3.0'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance' xmlns:-
domain='http://registry.denic.de/domain/3.0' xmlns:dnsentry-
='http://registry.denic.de/dnsentry/3.0'>
 <domain:create>
  <domain:handle>beispiel-fünf.de</domain:handle>
  <domain:ace> xn--beispiel-fnf-mlb.de</domain:ace>
  <domain:contact role='holder'>DENIC-99990-BSP</domain:contact>
  <dnsentry:dnsentry xsi:type="dnsentry:NS">
   <dnsentry:owner>xn--beispiel-fnf-mlb.de</dnsentry:owner>
   <dnsentry:rdata>
    <dnsentry:nameserver>ns1.provider.de</dnsentry:nameserver>
   </dnsentry:rdata>
  </dnsentry:dnsentry>
  <dnsentry:dnsentry xsi:type="dnsentry:NS">
   <dnsentry:owner>xn--beispiel-fnf-mlb.de</dnsentry:owner>
   <dnsentry:rdata>
    <dnsentry:nameserver>ns1.marco-testdns1.de</dnsentry:nameserver>
    <dnsentry:address>192.0.2.1</dnsentry:address>
   </dnsentry:rdata>
  </dnsentry:dnsentry>
  <dnsentry:dnsentry xsi:type="dnsentry:DNSKEY">
   <dnsentry:owner>denic-itstt-11864785.de.</dnsentry:owner>
   <dnsentry:rdata>
    <dnsentry:flags>257</dnsentry:flags>
    <dnsentry:protocol>3</dnsentry:protocol>
    <dnsentry:algorithm>5</dnsentry:algorithm>
    <dnsentry:pub-
licKey>AwEAAdDECa-
jHaTjfSoNTY58WcBah1BxPKVIHBz4IfLjfqMvi-
um4lgKtKZLe97DgJ5/NQrNEGGQmr6fKvUj67cfrZ
UojZ2cGR-
izVhgkOqZ9s-
caTVXNuXLM5Tw7VWOVIceeXAuuH2mPIiEV6MhJYUsW6d-
vmNsJ4XwCgN-
groAmXhoMEiWEjBB+wjYZQ5GtZHBFKVXACSWTiCt-
ddHcueOeSVPi5WH94Vlub-
hHfiytNPZLrOb-
hUCHT6k0tNE6phLoHnXWU+6vpsYpz6GhMw/R9BFxW5PdPFIWBgoWk2/XFVRSKG9Lr61-
b2z1R126xeUwvw46RVy3h anV3vNO7LM5H niqaYclBbhk=</dnsentry:publicKey>
   </dnsentry:rdata>
  </dnsentry:dnsentry>
 </domain:create>
</registry-request>
```

Special Features

- **CREATE** If a domain is registered successfully but the connection fails (the domain is registered with the status “failed”), the DS records calculated from the Dnskey records are not taken into account when generating the zone.
- **UPDATE / CHHOLDER** If there are no DNSKEY records for a domain, these can be entered using an UPDATE or CHHOLDER.

  If one or more Dnskey entries already exist for a domain and the keyword “Dns-key:” is not present or is empty in a domain UPDATE or domain CHHOLDER order, the order will be executed and the original value of “Dnskey:” will be deleted.
- **CHPROV**, See the “Change of operator” documentation
- **TRANSIT** Any DNSkey entries are deleted for TRANSIT requests (regardless of whether the value of “Disconnect:” is “true” or “false”).
- **DELETE** For DELETE requests, the DS records generated from the Dnskey entries are also removed from the.de zone.

Nameserver Predelegation Check

For more information, see the documentation "Nameserver Predelegation Check" and at https://nast.denic.de.

Network Byte Order

Introduction

In computing, byte order or endianness is the way simple numeral values are stored. It names in particular the storing order of integers in the main memory. The applicable storage format must be defined if the number of bits required to encode the numeral to be stored is larger than the number of bits available in the smallest addressable unit. Normally, the smallest addressable unit is one byte. If more than one byte is needed for storing, the numeral is then stored in several

bytes with the memory addresses stated in immediate sequence. Whilst cross-producer standards have been established with many other types of storage organization, two variants persist for byte order.

Big Endian

In the case of Big Endian, the byte with the bits of the largest value (i.e. the most significant values) are stored first, i.e. at the lowest memory address. Generally speaking, the term Big Endian defines that data of the largest unit is to be stated first. It can be compared to the German manner of stating the time: hours:minutes:seconds.

Little Endian

In the case of Little Endian in contrast, the byte with the bits of the lowest value (i.e. the least significant values) are stored at the lowest memory address. Here, the data of the smallest unit is to be stated first. It can be compared to the German manner of stating a date: day:month:year.

Converting Big Endian to Little Endian

Since it is occasionally necessary to convert the one format to the other (especially in network programming) the commonly used programming languages include special functions for that. "C" for example offers the functions "htonl()" and "ntohl", Perl uses "pack" and "unpack" with the "N" parameter. The network byte order must be given in binary.

Reading and Sending Data

Example in C for reading data

```
/*!\brief Read answer from RRI server
*
* Use this function to read an answer from the RRI-Server
*
* @param ssl SSL context
* @return Pointer to a c-string, which contains the answer. The
memory used for this
* string is allocated with malloc and must be freed by the applic-
ation with free().
* If an error occured, the function returns NULL.
*/
char *rriRead(SSL *ssl)
{
    char *buf;
    int s,nl,size,rest;
    nl=0;
    size=0;
    rest=4;
    while (rest) {
        s=SSL_read(ssl,(char*)&nl+(4-rest),rest);
        if (s<=0) return (NULL);                  // Fehler auf-
getreten
        rest-=s;
    }
    size=ntohl(nl);
    buf=(char*)malloc(size+2);
    if (!buf) {
        // Not enough memory
        return (NULL);
    }
    rest=size;
    while (rest) {
        s=SSL_read(ssl,(char*)buf+(size-rest),rest);
        if (s<=0) {
           free(buf);
           return (0);                            // Fehler auf-
getreten
        }
       rest-=s;
    }
    buf[size]=0;
    return (buf);
}
```

Example in C for sending data

```
/*!\brief Send order to RRI server
*
* Use this function to send an order to the RRI server
*
* @param ssl SSL context
* @param order C-String, which contains the UTF-8 encoded order
* @return Returns the length of the order, if it was successful send
to the
* RRI server or 0, if an error occured.
*/
int rriSend(SSL *ssl, const char *order)
{
  int size,nl,s;
  size=(int)strlen(order);
  nl=htonl(size);
  // Send 4-Byte perfix with length of order in bytes
  s  =SSL_write(ssl,(char*)&nl,4);
  if (s!=4) {
      // Error
      return (0);
  }
  // Send order
  s=SSL_write(ssl,order,size);
  if (s!=size) {
      // Error
      return (0);
  }
  return (s);
}
```

Example in PHP for reading data

```
/*!\brief Read answer from RRI-Server
*
* \param $socket Socket handle
* \return Returns string with answer from RRI server or false, if an
error
* occured.
*/
function rriReadData($socket)
{
    // Step 1: read 4-Byte RRI-Header
    $nlen="";
    $rest=4;
    while ($rest) {
        $a=fread($socket,$rest);                // read answer
        $bytesread=strlen($a);
        $nlen.=$a;
        $rest-=$bytesread;
        if (feof($socket)) return false;
    }

    $len=rriUnpack($nlen);                 // convert bytes to local
order

    // Step 2: read payload
    $rest=$len;
    $answer="";
    while ($rest) {
        $a=fread($socket,$rest);                // read answer
        $bytesread=strlen($a);
        $answer.=$a;
        $rest-=$bytesread;
        if (feof($socket)) return false;
    }
    return $answer;
}
```

The “rriUnpack” function is called up within the “rriReadData” function. This is how it reads:

```
/*!\brief Converts network byte order to local byte order
*
* \param $value Integer in network byte order
* \return Returns the converted $value in local byte order
*/
function rriUnpack($value)
{
    $a[0]=ord($value[0]);
    $a[1]=ord($value[1]);
    $a[2]=ord($value[2]);
    $a[3]=ord($value[3]);
    return $a[3]+($a[2]<<8) + ($a[1]<<16) + ($a[0]<<24);
}
```

Example in PHP for sending data

```
/*!\brief Sends order to RRI-Server
*
* \param $socket Socket handle
* \param $order String, which should be send to the RRI-server
* \return Returns 1 on success, 0 on failure
*/
function rriSendData($socket, $order)
{
    $len=strlen($order);
    $nlen=pack("N",$len);           // Convert Bytes of len to net-
work order
    if (fwrite($socket,$nlen,4)<4) return 0;  // send length of
order to server
    if (fwrite($socket,$order,$len)<$len) return 0;  // send order
    return 1;
}
```

Example in Perl for reading data

```
# Read answer from RRI-Server
#
# Syntax: rriReadData(object socket)
# Parameter:
#    - socket: SSL connection established with rriConnect
#
# Returnvalue:
#    Returns string with answer from RRI server on success or
"undef", if an error
#    occured.
sub rriReadData {
        my $sock=shift;
        my ($head, $head2);
        my ($data, $data2);
        my $ret;
        $head="";
        $data="";

        # Step 1: read 4-byte RRI-header
        my $rest=4;
        while ( $rest ) {
            $ret=read $sock,$head2,$rest;
            if (! defined($ret)) {
                    return (undef);
            }
            $head.=$head2;
            $rest-=$ret;
        }
        my $len=unpack "N",$head;
        if ($len > 65535) {        # Should not happen, something
went wrong
                    return (undef);
        }

        # Step 2: read payload
        $rest=$len;
        while ( $rest ) {
            $ret=read $sock,$data2,$rest;
            if (! defined($ret)) {
                    return (undef);
            }
            $data.=$data2;
            $rest-=$ret;
        }
        return $data;
}
```

Example in Perl for sending data

```
# Sends order to RRI-Server
#
#
# Syntax: rriSendData(object socket, string order)
# Parameter:
#    - socket: SSL connection established with rriConnect
#    - order: String, which should be send to the RRI-server
#
# Returnvalue:
#    Returns 1 on success, 0 on failure
sub rriSendData {
        my ($sock,$data)=@_;
        my $len=length($data);                  # Length of data
        my $head=pack "N",$len;                 # convert to 4 byte
value in network byteorder
        return 0 if (!print $sock $head);       # send 4 byte header
        return 0 if (!print $sock $data);       # send payload
        return 1;
}
```

Example in Java for reading and sending data

```
package de.denic.rri.common;

import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.Serializable;
import java.io.UnsupportedEncodingException;
import java.net.InetSocketAddress;
import java.net.Socket;
import java.nio.ByteBuffer;
import java.nio.CharBuffer;
import java.nio.charset.CharacterCodingException;
import java.nio.charset.Charset;
import java.nio.charset.CharsetDecoder;

import org.apache.commons.lang.Validate;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;

public final class TcpProtocolFramingHandler implements RriConstants
{

  private static final Log log = LogFactory.getLog(TcpPro-
tocolFramingHandler.class);
  // Immutable and thread-safe instance:
  private static final Charset PROTOCOL_CHARACTER_SET = Char-
set.forName(NAME_OF_PROTOCOLS_CHARACTER_SET);

  // Instance is NOT thread-safe!
  private final CharsetDecoder protocolsCharsetDecoder = PROTOCOL_
CHARACTER_SET.newDecoder();
  private final Socket socket;

  public TcpProtocolFramingHandler(final Socket socket) throws
IllegalArgumentException {
    super();
    Validate.notNull(socket, "Missing socket");
    this.socket = socket;
```

```
  }

  public MessageDataWithOptionalExceptionValue nextFrame() throws
IOException {
    return nextFrame(-1); // Keine Prüfung der erlaubten Länge des
Frames
  }

  public MessageDataWithOptionalExceptionValue nextFrame(final int
maxAllowedPayloadSize) throws IOException {
    return doNextFrame(maxAllowedPayloadSize, socket.getInputStream
());
  }

  protected final MessageDataWithOptionalExceptionValue doNextFrame
(final int maxAllowedPayloadSize, final InputStream inputStream)
throws IOException, IllegalArgumentException {
    final long lengthOfFrameInBytes = evaluateLengthOfFrameInBytes
(inputStream);
    if (lengthOfFrameInBytes == 0) {
      return new MessageDataWithOptionalExceptionValue("", null);
    }

    if ((maxAllowedPayloadSize > 0) && (lengthOfFrameInBytes >
maxAllowedPayloadSize)) {
      return new MessageDataWithOptionalExceptionValue("[Message to
long]", new RriException(RriExceptionType.MESSAGE_TOO_LONG, new
Serializable[] {Long.toString(lengthOfFrameInBytes) }), true);
    }

    final int lengthOfFrameInBytesAsInt = (int)
lengthOfFrameInBytes;
    final byte[] messageAsByteArray = new byte
[lengthOfFrameInBytesAsInt];
    if (log.isInfoEnabled()) {
      log.info("Reading frame of " + lengthOfFrameInBytes + "
bytes");
    }
    int bytesReadForMessage = 0;
    while (bytesReadForMessage < lengthOfFrameInBytes) {
```

```
 3; i >= 0; i--) {
      final int byteRead = inputStream.read();
      if (byteRead < 0) {
        if (i == 3) {
          // No single byte reaches us
          throw new IOException("Input stream of connection is
empty: Connection seems to be closed");
        }

        throw new IOException("Missing four bytes representing
frame's length");
      }

      lengthOfFrameInBytes += byteRead << (8 * i);
    }
    if ((lengthOfFrameInBytes > Integer.MAX_VALUE) ||
(lengthOfFrameInBytes < 0)) {
      throw new IOException("The four bytes representing frame's
length denotes a number that cannot be handled: "
        + lengthOfFrameInBytes);
    }

    return lengthOfFrameInBytes;
  }

  public void putFrame(final String frame) throws IOException {
    doPutFrame(frame, socket.getOutputStream());
  }

  /**
  * Access modifier protected for testing purposes
  */
  protected final void doPutFrame(final String frame, final Out-
putStream outputStream)
  throws UnsupportedEncodingException, IOException {
   final byte[] frameAsUtf8Bytes = frame.getBytes(NAME_OF_PROTOCOLS_
CHARACTER_SET);
   final int lengthOfFrameInBytes = frameAsUtf8Bytes.length;
   final byte[] lengthEncodedAsBytes = new byte[] {(byte)
(lengthOfFrameInBytes >> 24),
      (byte) (lengthOfFrameInBytes >> 16), (byte)
(lengthOfFrameInBytes >> 8), (byte) lengthOfFrameInBytes };
   if (log.isInfoEnabled()) {
     log.info("Writing frame of " + lengthOfFrameInBytes + "
```

List of Message Codes

The list is sorted in ascending order by the message code.

> **Notice**
>
> For some error texts, a variable text is displayed in addition to the fixed text. This is intended to explain the problem in more detail. The position of the variable text is indicated by “[dynamic error text]”.

contact CREATE and contact UPDATE

Message code	Classification	Error text	Description
63200042402	ERROR	Keyword "Phone" must appear once and just once	If a Contact CREATE or Contact UPDATE contains no or more than one phone number, the request is rejected.
63200062603	ERROR	The value for keyword "Phone" has an invalid format	If a phone number in a Contact CREATE or Contact UPDATE is in the wrong format, the request is rejected.

domain CREATE

Classification	Message code	Error text	Description
WARNING	53000080014	Domain "Status" is "pendingCreate"	When the domain is handed over to the risk assessment without a final status (connect, serverHold), the domain status “pendingCreate” is set and a warning is issued.

domain CHHOLDER

Classification	Message code	Error text	Description
ERROR	63300085940	Missing phone for holder [Output of the handle(s)]	• A CHHOLDER domain request will fail if a **new** contact is used that does not have a phone number. • For Multiholder domains, all contacts must have a phone number. **Caution!** A domain CHHOLDER can continue to have a contact without a phone number as long as the holder does not change.

domain CHPROV

Classification	Message code	Error text	Description
ERROR	63300085940	Missing phone for holder [Output of the handle(s)]	• A domain CHPROV fails if a contact is used that has no phone number. • For Multiholder domains, all contacts must have a phone number.

domain RESTORE

Classification	Message code	Error text	Description
ERROR	63300085940	Missing phone for holder [Output of the handle(s)]	• A domain restore with an AuthInfo2 will fail if a contact is used that does not have a phone number. • When restoring with an AuthInfo2, all contacts of a multiholder domain must have a phone number. **Caution!** A domain restore without AuthInfo is also possible with contacts without phone numbers.

Message Code	Message Text
11300088725	AuthInfo deleted
11300088726	AuthInfo letter could not be delivered
13000000011	Request was processed in test environment – not valid in real world
13000000022	Used quota
13350000001	Domain bears a DISPUTE entry and can therefore not be restored
16000063119	DNS data was checked and accepted
16330088732	An action determines the removal of the AuthInfo
16350000031	Domain has been deleted [request-reference DENIC#:<Ticket-ID>]
33000000022	Used quota

Message Code	Message Text
33300102900	The provided DNS data are not acceptable. Domain "Status" is "failed".
33300102912	Predelegation Check warning
46000108827	Secure Entry Point flag should be set in DNSKEY
53000000008	Daily quota exceeded
53000000012	Concurrent write access to the same business object is prohibited [dynamic error text]
53000020002	Unauthorized request
53000020013	Verification of signature failed
53000020019	Signature of request is correct but too old
53000060009	At least one Contact does not exist

Message Code	Message Text
53000080002	This value cannot be deleted
53000080011	The request cannot be carried out because of an active DISPUTE
53000080012	Domain “Status” is “failed”
53300062003	The value for keyword "Domain" has an invalid format [dynamic error text]
53300081600	Within a Contact “Email” is missing
53300100021	Technical Predelegation Check Problem – please try later
53300102902	Required glue record is missing [dynamic error text]
53300102912	Nameserver error [dynamic error text]
53300102920	Occurrences of keyword “Nserver” out of range [min: 2, max: 13]

Message Code	Message Text
53300103022	Missing mail exchange preference of DNS entry
53370080005	Required change provider request data is missing
53380062206	Mismatch between values for “Domain-ace” and “Domain”
56000000013	Technical error
56000020021	Access control limit exceeded
56000088709	Specified AuthInfo2 does not exist
56300108812	The provided DNS data are not acceptable
56310082011	Domain "Status" is "redemptionPeriod"
58000000002	Text part has not been encoded properly

Message Code	Message Text
58000000027	Mailheader is incomplete
63000000006	The request has an invalid key/value pair format
63000000007	Request shows semantics error [dynamic error text]
63000000012	Unknown Error
63000040110	Keyword “Action” cannot be recognized
63000040410	Keyword “Version” cannot be recognized
63000060003	An indicated Contact is invalid [repetition of the invalid handle]
63000060008	The value contains one or more invalid characters [dynamic error text]
63000060015	The value is too long [dynamic error text]

Message Code	Message Text
63000060102	Inadmissible value for keyword “Action”
63000060402	Inadmissible value for keyword “Version”
63000060504	“CTID” has been used before [dynamic error text]
63000060508	Inadmissible value for keyword "Ctid"
63000080009	Request type is missing, either keyword “Domain” or “Handle”
63100020005	Number of failed logins exceeds allowed maximum
63100040202	Keyword “User” must appear once and just once
63100040302	Keyword “Password” must appear once and just once
63100064002	Authorization failure

Message Code	Message Text
63200020603	This Contact is administered by another provider [dynamic error text]
63200040602	Keyword “Handle” or “Domain” must appear once and just once
63200040610	Keyword “Handle” cannot be recognized
63200040702	Keyword “Type” must appear once and just once
63200040710	Keyword “Type” cannot be recognized
63200040802	Keyword “Name” must appear once and just once
63200040810	Keyword “Name” cannot be recognized
63200041010	Keyword “Address” cannot be recognized
63200041102	Keyword “PostalCode” must appear once and just once

Message Code	Message Text
63200041110	Keyword “PostalCode” cannot be recognized
63200041202	Keyword “City” must appear once and just once
63200041210	Keyword “City” cannot be recognized
63200041302	Keyword “CountryCode” must appear once and just once
63200041310	Keyword “CountryCode” cannot be recognized
63200041610	Keyword "Email" cannot be recognized
63200041802	Keyword “Disclose” can appear once and just once
63200060702	Inadmissible value for keyword “Type”
63200061302	The value for “CountryCode” is not a valid ISO code

Message Code	Message Text
63200061603	The value for keyword “Email” has an invalid format
63200061708	Invalid character(s) in “Sip” value
63200061802	Inadmissible value for keyword “Disclose”
63200080108	This value of “Action” is not permitted here
63200080900	Keyword “Organisation” invalid with “Type” “ORG”
63220080701	The value of “Type” cannot be changed with an UPDATE
63220080801	The value of “Name” cannot be changed with an UPDATE
63300022003	This domain is administered by another provider [dynamic error text]
63300042100	Keyword “Domain-ace” cannot be recognized

Message Code	Message Text
63300042102	Keyword “Domain-ace” can appear once and just once
63300043003	Occurences of keyword "Nsentry" out of range [dynamic error text]
63300043100	Keyword “Nserver” or “Nsentry” cannot be recognized
63300060005	Contact occurs more than once in same role
63310062008	Inadmissible value for keyword "RegAcc" [Repeat of the user entry]
63300062009	Domain doesn’t exist [repetition of the non-existent domain]
63300062010	RegAcc doesn't exist [Repeat of the RegAcc]
63300062905	Duplicate value(s) for “Nserver”
63300063002	Preference values for MX-RR are out of range (0-999)

Message Code	Message Text
63300063003	Incorrect value(s) for “Nsentry”
63300063005	Duplicate value for “Nsentry” record is present
63300063006	“Nsentry” has invalid owner
63300063011	Nserver has invalid owner [dynamic error text]
63300063016	At least 2 MX entries have the identical preference
63300063102	Unknown type of DNS data
63300063103	Illegal type of DNS data
63300080010	Required domain data is missing
63300080013	Too many Contacts are indicated

Message Code	Message Text
63300080700	Contact with wrong value for “Type”
63300083005	Value for “Nsentry” contains unknown host
63300102913	Incorrect value(s) for “Nserver” [dynamic error text]
63300103005	No other MX resource record is allowed in combination with "Null MX"
63310062004	Domain already exists
63310062007	Inadmissible value for keyword "Domain" [dynamic error text]
63360043403	Keyword “Disconnect” can appear once and just once
63360063402	Inadmissible value for keyword “Disconnect”
63360082531	For TRANSIT, a Holder Handle is mandatory

Message Code	Message Text
63370048700	Keyword “AuthInfo” cannot be recognized [dynamic error text]
63370088701	AuthInfo does not match [dynamic error text]
63380063209	Illegal operation on own domain
63391048502	Keyword “AuthInfoHash” must appear once and just once
63391048602	Keyword “AuthInfoExpire” can appear once and just once
63391068503	The value for keyword "AuthInfoHash" has an invalid format
63391068603	The value for keyword "AuthInfoExpire" has an invalid format
63391068704	AuthInfo1 already exists
63391082031	Domain has Description

Message Code	Message Text
63392068709	AuthInfo1 does not exist
63393068704	AuthInfo2 already exists
63450043902	Keyword “Recursive” can appear once and just once
63450063902	Inadmissible value for keyword “Recursive”
63600063502	Inadmissible value for keyword “MsgType”
63650063607	This message is not the oldest
63650063609	MsgId does not exist
63900042502	Keyword “Holder” must appear once and just once
66000020020	Invalid Signature Algorithm

Message Code	Message Text
66300048905	Invalid DNS Data (Nsentry, Dnskey)
66300063003	Missing mail exchange host
66300063006	“Nsentry” has invalid owner
66300063023	AAAA resource record requires corresponding A resource record
66300068803	The value for keyword "Dnskey" has an invalid format
66300068807	“Dnskey” invalid
66310082014	Domain is not in Redemption Grace Period
73100000005	The maximum number of permitted login is reached
76000000001	Registry Service temporarily not available

Message Code	Message Text
76000000002	Timeout while processing request
83000000000	Request message too long [dynamic error text]
83000000002	The encoding specified is not supported
83000000004	Validation of XML request against Schema failed
83000000005	Validation of URI-Template failed [„empty URI-Template“]
83000000005	Validation of URI-Template failed [URI-Template too long: max length allowed is 1024]
83000000005	Validation of URI-Template failed [invalid URI scheme (dynamic error text]
83000000005	Validation of URI-Template failed [invalid URI-Template variable(s) \[(dynamic error text)\]]
83000000005	Validation of URI-Template failed [The variable name (dynamic error text) contains invalid characters]

Message Code	Message Text
83000000010	Login failed

List of Message Codes for Verification

Message code	Classification	Error text	Description
63450064002	ERROR	One of the indicated values is not valid for this contact type [VerificationInformation]	A verification record was specified in a “REQUEST” contact. Verification data can only be specified for “PERSON” or “ORG” contacts.
63200062603	ERROR	Incomplete verification information [missing keyword]	A verification record is incomplete. The placeholder “keyword” contains the keyword of the missing line. **Example** If “verificationMethod” is missing, the error message will look like this: 63200041860 - Incomplete verification information [missing verificationMethod]
63200041860	ERROR	Incomplete verification information [missing VerificationInformation]	Since the introductory header “[VerificationInformation]” of the verification record is missing, the keywords are not recognized.

Message code	Classification	Error text	Description
63200041861	ERROR	Inadmissible value in verification information [dynamic error text]	The verification record contains at least one invalid value, e.g.: VerifiedClaim: noclaim The corresponding error message is : 63200041861 - Invalid value in verification information [verifiedClaim 'noclaim' is invalid] The placeholders “keyword” and “value” contain the corresponding keyword and the invalid value.
63200041862	ERROR	Occurrences of verification information out of range	More than 5 verification records were transmitted in one contact.

Message code	Classification	Error text	Description

63200041863	ERROR	Conflicting verification information [veri-ficationReference 'value']	**Notice**
			1st example of error message 63200041863
			
			Different verification information has been provided for the claim “name”. In the example, the value for “Veri-ficationResult” is different.
			
			The verification records are specified in the error mes-sage using their “VerificationReference” values. The “value” is replaced with the reference number. The "value" only appears once in this example because it is identical in both verification records.
			
			Example
			
			`1 [VerificationInformation]`
			`2 VerifiedClaim: name`

Message code	Classification	Error text	Description
			3 VerificationResult: success 4 VerificationReference: abc#111 5 VerificationTimestamp: 2024-04-01T14:58:30+02:00 6 VerificationEvidence: idcard 7 VerificationMethod: auth 8 TrustFramework: de_denic 9 10 [VerificationInformation] 11 VerifiedClaim: name 12 VerificationResult: failed 13 VerificationReference: abc#111 14 VerificationTimestamp: 2024-04-01T14:58:30+02:00 15 VerificationEvidence: idcard 16 VerificationMethod: auth 17 TrustFramework: de_denic **Error Message** 63200041863 - Conflicting verification information [verificationReference 'abc#111']

Message code	Classification	Error text	Description

63200041863	ERROR	Conflicting verification information [veri-ficationReference 'value 1' 'value 2']	**Notice**
			2nd example of error message 63200041863
			
			Different verification information has been provided for the claim “name”. In the example, the values for “Veri-ficationResult”, “VerificationReference” and “Veri-ficationEvidence” are different.
			
			The error message specifies the verification records using their “VerificationReference” values. The place-holders for the values (“value 1”, "value 2”) are replaced with the reference numbers.
			
			Example
			
			`[VerificationInformation]`
			`VerifiedClaim: name`
			`VerificationResult: success`
			`VerificationReference: abc#111`

Message code	Classification	Error text	Description
			5 VerificationTimestamp: 2024-04-01T14:58:30+02:00 6 VerificationEvidence: idcard 7 VerificationMethod: auth 8 TrustFramework: de_denic 9 10 [VerificationInformation] 11 VerifiedClaim: name 12 VerificationResult: failed 13 VerificationReference: abc#112 14 VerificationTimestamp: 2024-04-11T14:58:30+02:00 15 VerificationEvidence: drivers_license 16 VerificationMethod: auth 17 TrustFramework: de_denic **Error Message** 63200041863 - Conflicting verification information [verificationReference 'abc#111' 'abc#112']

Message code	Classification	Error text	Description
63200041864	ERROR	Duplicate verification information	At least two identical blocks were transmitted for a contact in one order.

Message code	Classification	Error text	Description

63200041865	ERROR	Duplicate value in verification information [Schlüsselwort 'name', 'address' oder 'email']	**Notice**
			1st example of error message 63200041865
			
			A value was specified twice for a keyword in a verification data set.
			
			Example
			
			`[VerificationInformation]`
			`VerifiedClaim: name`
			`VerifiedClaim: name`
			`VerificationResult: success`
			`VerificationReference: abc#444`
			`VerificationTimestamp: 2024-04-02T14:58:30+02:00`
			`VerificationEvidence: idcard`
			`VerificationMethod: auth`
			`TrustFramework: de_denic`
			
			Error Message

Message code	Classification	Error text	Description
			63200041865 - Duplicate value in verification information [VerifiedClaim 'name']

Message code	Classification	Error text	Description
63200041865	ERROR	Duplicate value in verification information [(Schlüsselwort) '(Wert "name", "address" oder "email")']	**Notice** 2nd example of error message 63200041865 In the example, the claim “name” appears in two blocks. The blocks are identical and therefore “name” is also duplicated, as the values for • VerificationResult, • VerificationReference, • VerificationTimestamp, • VerificationEvidence, • VerificationMethod and • TrustFramework are also identical. **Example**

Message code	Classification	Error text	Description

			<pre>1 [VerificationInformation]
			2 VerifiedClaim: name
			3 VerifiedClaim: address
			4 VerificationResult: success
			5 VerificationReference: abc#444
			6 VerificationTimestamp: 2024-04-02T14:58:30+02:00
			7 VerificationEvidence: idcard
			8 VerificationMethod: auth
			9 TrustFramework: de_denic
			10
			11 [VerificationInformation]
			12 VerifiedClaim: name
			13 VerifiedClaim: email
			14 VerificationResult: success
			15 VerificationReference: abc#444
			16 VerificationTimestamp: 2024-04-02T14:58:30+02:00
			17 VerificationEvidence: idcard
			18 VerificationMethod: auth
			19 TrustFramework: de_denic</pre>
			Error Message
			63200041865 - Duplicate value in verification

Message code	Classification	Error text	Description
			`information [VerifiedClaim 'name']`

Message code	Classification	Error text	Description

63200041866	ERROR	Duplicate key in verification information [keyword]	A keyword was specified twice in a verification data set.
			
			Caution!
			Please do not confuse this error with error 63200041865, in which the value occurs multiple times. In error 63200041866, the keywords are identical and the values are different.
			
			Example
			
			`# Key VerificationReference doppelt`
			`[VerificationInformation]`
			`VerifiedClaim: name`
			`VerificationResult: success`
			`VerificationReference: abc#444`
			`VerificationReference: DEF#445`
			`VerificationTimestamp: 2024-04-02T14:58:30+02:00`
			`VerificationEvidence: idcard`
			`VerificationMethod: auth`
			`TrustFramework: de_denic`

Message code	Classification	Error text	Description
			Error Message 63200041866 - Duplicate key in verification information [verificationReference]
63200041867	ERROR	Invalid combination of verification information [(Comparison of the impermissible combinations)]	Invalid combination of verification information for 'VerificationEvidence' and 'VerificationMethod' **Error Message** Invalid combination of verification information [verificationEvidence 'bank_statement' not valid for verificationMethod 'auth']

Link Collection

Public Websites

- DENIC Instructions and DENIC documentation https://docs.denic.de
- Mailing lists https://www.denic.de/en/service/mailing-lists
- NAST http://nast.denic.de

Member Websites

- Password encryption tools https://member.secure.denic.de/en/service/tools/hash-generator
- Download center (24-hour agreement, service standards and much more) https://member.secure.denic.de/en/service/downloads-and-archives
- Accounting data https://member.secure.denic.de/en/service/financial-issues/accounting-data
- DENIC's public keys https://member.secure.denic.de/en/contact/pgp-keys
- RRI web interfaces https://member.secure.denic.de/en/service/tools/rri-web-client
- Member mailing lists https://member.secure.denic.de/en/service/member-mailing-lists

External Links

- XML Schema data types https://www.w3.org/TR/xmlschema-2
- Unicode 3.1 http://www.unicode.org/reports/tr27/tr27-4.html
- ISO 3166-2:2007 Codes for the representaon of names of countries and their subdivisions – Part 2: Country subdivision code https://www.iso.org/iso-3166-country-codes.html
- E.123 Notation for national and international telephone numbers, e-mail addresses and Web addresses http://www.itu.int/rec/T-REC-E.123/en

- IANA Ipv6 Global Unicast Address Assignments http://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xml
- IANA Ipv6 Special Purpose Address Registry http://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xml

Relevant RFCs

- RFC791 - INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION https://datatracker.ietf.org/doc/html/rfc791
- RFC793 – TRANSMISSION CONTROL PROTOCOL https://datatracker.ietf.org/doc/html/rfc793
- RFC1034 – DOMAIN NAMES – CONCEPTS AND FACILITIES https://datatracker.ietf.org/doc/html/rfc1034
- RFC1035 – DOMAIN NAMES – IMPLEMENTATION AND SPECIFICATION https://datatracker.ietf.org/doc/html/rfc1035
- RFC1123 – Requirements for Internet Hosts – Application and Support https://datatracker.ietf.org/doc/html/rfc1123
- RFC1305 – Network Time Protocol (Version 3) https://datatracker.ietf.org/doc/html/rfc1305
- RFC1321 – The MD5 Message-Digest Algorithm https://datatracker.ietf.org/doc/html/rfc1321
- RFC2104 – HMAC: Keyed-Hashing for Message Authentication https://datatracker.ietf.org/doc/html/rfc2104
- RFC2181 – Clarification to the DNS Specification https://datatracker.ietf.org/doc/html/rfc2181
- RFC2308 – Negative Caching of DNS Queries (DNS NCACHE) https://datatracker.ietf.org/doc/html/rfc2308

- RFC2396 – Uniform Resource Identifiers (URI): Generic Syntax https://datatracker.ietf.org/doc/html/rfc2396
- RFC2536 – DSA KEYs and SIGs in the Domain Name System (DNS) https://datatracker.ietf.org/doc/html/rfc2536
- RFC2960 – Stream Control Transmission Protocol https://datatracker.ietf.org/doc/html/rfc2960
- RFC3023 – XML Media Types https://datatracker.ietf.org/doc/html/rfc3023
- RFC3080 – The Blocks Extensible Exchange Protocol Core https://datatracker.ietf.org/doc/html/rfc3080
- RFC3110 – The RSA/SHA-1 SIGs and RSA KEYs in the Domain Name System (DNS) https://datatracker.ietf.org/doc/html/rfc3110
- RFC3156 – MIME Security with OpenPGP https://datatracker.ietf.org/doc/html/rfc3156
- RFC3261 – SIP: Session Initiation Protocol https://datatracker.ietf.org/doc/html/rfc3261
- RFC3658 - Delegation Signer (DS) Resource Record (RR) https://datatracker.ietf.org/doc/html/rfc3658
- RFC4034 - Resource Records for the DNS Security Extensions https://datatracker.ietf.org/doc/html/rfc4034
- RFC4291 - IP Version 6 Addressing Architecture https://datatracker.ietf.org/doc/html/rfc4291
- RFC4641 – DNSSEC Operational Practices https://datatracker.ietf.org/doc/html/rfc4641
- RFC4648 – The Base16, Base32, and Base64 Data Encodings https://datatracker.ietf.org/doc/html/rfc4648
- RFC4697 – Observed DNS Resolution Misbehaviour https://datatracker.ietf.org/doc/html/rfc4697
- RFC4880 – OpenPGP Message Format https://datatracker.ietf.org/doc/html/rfc4880

- RFC5011 - Automated Updates of DNS Security (DNSSEC) Trust Anchors https://datatracker.ietf.org/doc/html/rfc5011
- RFC5246 – The Transport Layer Security (TLS) Protocol Version 1.2 https://datatracker.ietf.org/doc/html/rfc5246
- RFC5322 – Internet Message Format https://datatracker.ietf.org/doc/html/rfc5322
- RFC5734 – Extensible Provisioning Protocol (EPP) https://datatracker.ietf.org/doc/html/rfc5734
- RFC7505 - A "Null MX" No Service Resource Record for Domains That Accept No Mail https://datatracker.ietf.org/doc/html/rfc7505
- RFC6125 - Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS) https://datatracker.ietf.org/doc/html/rfc6125