Table of Contents

List of Tables

Chapter 1. Constraints on EPP elements

This chapter describes various constraints, which must be fulfilled, when submitting EPP command to server. The description is intended to be used by either an implementor of EPP client software or enduser of such a client. In order to ease navigation in document, it is structured in sections which bear names of EPP commands accepted by a server. Each such section describes an input parameters, their types and other constraints, the same description for output parameters and possible return codes. We start by EPP session management commands (login, logout), then follow query commands (check, info, poll), transformation commands (create, delete, renew, transfer, update) and finally extension commands (sendAuthInfo, test, creditInfo) and subgroup of extension commands info-commands (listDomains, listContacts, listNssets, listKeysets, domainsByNsset, domainsByKeyset, domainsByContact, nssetsByContact, nssetsByNs, keysetsByContact, getResults). We give a quick overview of used XML schema data types in apendix. For more detailed description of XML schema data types see XML Schema W3C specification [http://www.w3.org/XML/Schema#dev]. Majority of the document's content is based on definition of EPP protocol in XML schemas, therefore if in doubt about correctness of information presented in this paper, always consult EPP schemas.

Names of parameters are derived from names of XML tags, which surround the value. Parameters, which are not relevant from user's point of view, are omitted (e.g. version of EPP protocol).

1.1. Login and Logout

login logs a client into the system. Said otherwise, it creates a session. And does not have output parameters. *logout* terminates session established by *login*. *logout* does not have any i/o parameters.

Current limit on number of openned connections by one client is one hundered. If client is idle for 300 seconds, he is automatically logged out.

Table 1.1. login input parameters

Element	Occurence	Valid value	Description	Note
clID	1	clID	Login name	
pw	1	pw	Password	
newPW	0..1	pw	New password for future logins.	
lang	1	lang	Preferred language of messages from server.	

Possible return codes for login are rc1000, rc1000, rc2200, rc2502 and for logout rc1500.

1.2. Check

check command checks if handle is available for registration or not. In case that handle cannot be registered, server returns a reason.

Table 1.2. check domain input parameters

Element	Occurence	Valid value	Description	Note
name	1..n	domain	Handle of domain.	

Table 1.3. check domain output parameters

Element	Occurence	Valid value	Description	Note
name	1..n	domain	Handle of domain.	
available	1..n	boolean	True if the handle can be registered, false if not.	
reason	0..n	Check message	Reason why the handle cannot be registered.	

Table 1.4. check contact input parameters

Element	Occurence	Valid value	Description	Note
id	1..n	contactID	Handle of contact.	

Table 1.5. check contact output parameters

Element	Occurence	Valid value	Description	Note
id	1..n	contactID	Handle of contact.	
available	1..n	boolean	True if the handle can be registered, false if not.	
reason	0..n	Check message	Reason why the handle cannot be registered.	

Table 1.6. check nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1..n	nssetID	Handle of nsset.	

Table 1.7. check nsset output parameters

Element	Occurence	Valid value	Description	Note
id	1..n	nssetID	Handle of nsset.	
available	1..n	boolean	True if the handle can be registered, false if not.	
reason	0..n	Check message	Reason why the handle cannot be registered.	

Table 1.8. check keyset input parameters

Element	Occurence	Valid value	Description	Note
id	1..n	keysetID	Handle of keyset.	

Table 1.9. check keyset output parameters

Element	Occurence	Valid value	Description	Note
id	1..n	nssetID	Handle of nsset.	
available	1..n	boolean	True if the handle can be registered, false if not.	
reason	0..n	Check message	Reason why the handle cannot be registered.	

Expected return code is rc1000 or an error code in case of failure.

1.3. Info

info retrieves information about an object.

Table 1.10. info domain input parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Handle of domain.	

Table 1.11. info domain output parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Handle of domain	
roid	1	ROID	Repository object identifier	
status	0..9	domain status	Status of domain	
registrant	0..1	contactID	Registrant	Registrant is always displayed except when the domain is of type ENUM and the client does not own the domain.
admin	0..n	contactID	Administrative contact for domain	
nsset	0..1	nssetID	Handle of nsset, which gathers nameservers, on which the domain is delegated.	
clID	1	clID	Owner of domain (a registrar)	
crID	0..1	clID	Creator of domain (a registrar)	
crDate	0..1	Date and time	Date of domain creation	
upID	0..1	clID	ID of last registrar who did update of domain	
upDate	0..1	Date and time	Date of last domain update	
exDate	0..1	Date	Date of domain expiration	

Element	Occurence	Valid value	Description	Note
trDate	0..1	Date and time	Date of last transfer of domain	
authInfo	0..1	authInfo	Authorization information for domain.	This item is displayed only if client is owner of the domain.
tempcontact	0..n	contactID	Temporary contact of domain.	This element was introduced to ease transition from old system to the new one and hopefully will fade away in future.
valExDate	0..1	Date	Expiration of validation of ENUM domain	This information is there only for enum domains.
publish	0..1	boolean	Flag for publishing ENUM number and associated contact in public directory	This information is valid only for enum domains.

Table 1.12. info contact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	Handle of contact.	

Table 1.13. info contact output parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	Handle of contact	
roid	1	ROID	Repository object identifier	
status	0..4	contact status	Status of contact	
name	1	postal line	Name of contact	
org	0..1	optional postal line	Organization of contact	
street	1..3	postal line	Street	

Element	Occurence	Valid value	Description	Note
city	1	postal line	City	
sp	0..1	optional postal line	State or province	
pc	1	postal code	Postal code	
cc	1	country code	Country code	
voice	0..1	telephone number	Telephone number	
fax	0..1	telephone number	Fax number	
email	1	emailCommaList-Type	Email address	
clID	1	clID	Owner of contact (a registrar)	
crID	0..1	clID	Creator of contact (a registrar)	
crDate	0..1	Date and time	Date of contact creation	
upID	0..1	clID	ID of last registrar who did update of contact	
upDate	0..1	Date and time	Date of last contact update	
trDate	0..1	Date and time	Date of last transfer of contact	
authInfo	0..1	authInfo	Authorization information for a contact	
disclose	0..n	disclose	The elements inside disclose element decide, which contact attributes are public or private.	
vat	0..1	VAT	Value added tax ID	
ident	0..1	ident	Unique identificator	Type of identification is chosen by attribute *type*.
notifyEmail	0..1	emailCommaList-Type	Email address of contact object	

Element	Occurence	Valid value	Description	Note
			where notifications are sent	

Table 1.14. info nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	Handle of contact or nsset	

Table 1.15. info nsset output parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of nsset	
roid	1	ROID	Repository object identifier	
status	0..4	nsset status	Status of nsset	
clID	1	clID	Owner of nsset (a registrar)	
crID	0..1	clID	Creator of nsset (a registrar)	
crDate	0..1	Date and time	Date of nsset creation	
upID	0..1	clID	ID of last registrar who did update of nsset	
upDate	0..1	Date and time	Date of last nsset update	
trDate	0..1	Date and time	Date of last transfer of nsset	
authInfo	1	authInfo	Authorization information for a nsset	
name	0..10	nameserver	FQDN of nameserver in nsset	
addr	0..n	IP address	IP address of nameserver in nsset	

Element	Occurence	Valid value	Description	Note
tech	1..n	contactID	Technical contact for nsset object	
reportlevel	1	Nsset level	Level of technical checks performed on nsset. The higher the level is, the more detailed tests are executed. Zero means no technical test will be executed at all.	

Table 1.16. info keyset input parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	Handle of contact or keyset	

Table 1.17. info keyset output parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	ID of keyset	
roid	1	ROID	Repository object identifier	
status	0..4	keyset status	Status of keyset	
clID	1	clID	Owner of keyset (a registrar)	
crID	0..1	clID	Creator of keyset (a registrar)	
crDate	0..1	Date and time	Date of keyset creation	
upID	0..1	clID	ID of last registrar who did update of keyset	
upDate	0..1	Date and time	Date of last keyset update	
trDate	0..1	Date and time	Date of last transfer of keyset	

Element	Occurence	Valid value	Description	Note
authInfo	1	authInfo	Authorization information for a keyset	
dnskey	0..10	dnskeyT	DNS key record	
tech	1..n	contactID	Technical contact for keyset object	

Expected return code is rc1000 or an error code in case of failure.

1.4. Poll

There are two poll commands: *poll request* (abbreviated to *poll req* in protocol) and *poll acknowledge* (abbreviated to *poll ack* in protocol). *Poll req* does not have any input parameters.

Table 1.18. poll acknowledge input parameters

Element	Occurence	Valid value	Description	Note
msgID	1	msgID	ID of message which is going to be acknowledged.	

Table 1.19. poll request output parameters

Element	Occurence	Valid value	Description	Note
qDate	0..1	Date and time	Date of message insertion in a message queue	
msg	0..1	Poll message	Message in XML format	See section about format of poll messages.
count	1	count	Number of unacknoledged messages in queue	
id	1	msgID	ID of first unacknoledged message in queue	

Poll acknoledge returns rc1000 if successfull and poll request returns rc1300 or rc1301 if no error occurs.

1.4.1. Poll message format

Table 1.20. Poll message "Low credit"

Element	Occurence	Valid value	Description	
zone	1	zone	Name of zone	
limit	1	credit	Lower limit for the zone	
credit	1	credit	Account balance for the zone	

Table 1.21. Poll message "Domain transfer"

Element	Occurence	Valid value	Description	
name	1	domain	FQDN of transferred domain	
trDate	1	Date and time	Date of transfer	
clID	1	clID	New owner of domain	

Table 1.22. Poll message "Nsset transfer"

Element	Occurence	Valid value	Description	
id	1	nssetID	ID of transferred nsset	
trDate	1	Date and time	Date of transfer	
clID	1	clID	New owner of nsset	

Table 1.23. Poll message "Keyset transfer"

Element	Occurence	Valid value	Description	
id	1	keysetID	ID of transferred keyset	
trDate	1	Date and time	Date of transfer	
clID	1	clID	New owner of keyset	

Table 1.24. Poll message "Contact transfer"

Element	Occurence	Valid value	Description	
id	1	contactID	ID of transferred contact	
trDate	1	Date and time	Date of transfer	
clID	1	clID	New owner of contact	

Table 1.25. Poll message "Impending expiration of domain"

Element	Occurence	Valid value	Description	
name	1	domain	FQDN of domain, which will soon expire.	
exDate	1	Date	Date of domain expiration	

Table 1.26. Poll message "Domain expired"

Element	Occurence	Valid value	Description	
name	1	domain	FQDN of expired domain	
exDate	1	Date	Date of domain expiration	

Table 1.27. Poll message "Domain outage from zone"

Element	Occurence	Valid value	Description	
name	1	domain	FQDN of domain, which was outaged from zone.	
exDate	1	Date	Date of domain expiration	

Table 1.28. Poll message "Domain deletion"

Element	Occurence	Valid value	Description	
name	1	domain	FQDN of domain, which was deleted from register.	

Element	Occurence	Valid value	Description	
exDate	1	Date	Date of domain expiration	

Table 1.29. Poll message "Nsset deletion"

Element	Occurence	Valid value	Description	
id	1	nssetID	ID of nsset, which was deleted for inactivity.	

Table 1.30. Poll message "Keyset deletion"

Element	Occurence	Valid value	Description	
id	1	keysetID	ID of keyset, which was deleted for inactivity.	

Table 1.31. Poll message "Contact deletion"

Element	Occurence	Valid value	Description	
id	1	contactID	ID of contact, which was deleted for inactivity.	

Table 1.32. Poll message "Technical test result"

Element	Occurence	Valid value	Description	
cltestid	0..1	Transaction ID	Test ID, which is a copy of clTRID in test command.	
id	1	nssetID	ID of nsset, which was tested.	
name	0..n	domain	Name of extra domains tested with nsset	
testname	0..n	Test name	Name of technical test	
status	0..n	boolean	The test was passed if true and failed if false	

Element	Occurence	Valid value	Description	
note	0..n	Note	Accompanying comment	

1.5. Create

create creates new object in repository and does not have any output parameters.

Table 1.33. create domain input parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Name of a domain to be created	
period	0..1	period	Registration period	Unit of time (year or month) is specified by attribute *unit*. If period is not specified, the default one is used.
nsset	0..1	nssetID	Identifier of nsset associated with a domain	
keyset	0..1	keysetID	Identifier of keyset associated with a domain	
registrant	1	contactID	Registrant's handle	
admin	0..n	contactID	Administration contact	
authInfo	0..1	authInfo	Authorization information for a domain.	If authorization information is empty string, it is generated by a server.
valExDate	0..1	Date	Expiration of validation of ENUM domain	This information is valid only for enum domains.
publish	0..1	boolean	Flag for publishing ENUM number and associat-	This information is valid only for enum domains.

Element	Occurence	Valid value	Description	Note
			ed contact in public directory	

Table 1.34. create contact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of contact to be created.	
name	1	postal line	Name of contact	
org	0..1	optional postal line	Organization	
street	0..3	optional postal line	Street (up to three elements)	
city	1	postal line	City	
sp	0..1	optional postal line	State or province	
pc	1	postal code	Postal code	
cc	1	country code	Country code	
voice	0..1	telephone number	Telephone number	
fax	0..1	telephone number	Fax number	
email	1	emailCommaList-Type	Email address	
authInfo	0..1	authInfo	Authorization information for a contact.	If authorization information is empty string, it is generated by a server.
disclose	0..n	disclose	The elements inside disclose element decide, which contact attributes are public or private.	
vat	0..1	VAT	Value added tax ID	
ident	0..1	ident	Unique identificator of contact object	Type of identification is chosen by attribute *type*.
notifyEmail	0..1	emailCommaList-Type	Email address of contact object	

Element	Occurence	Valid value	Description	Note
			where notifications are send.	

Table 1.35. create nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of nsset to be created	
name	2..10	nameserver	FQDN of nameserver	
addr	0..n	IP address	IP address of nameserver	Address should be provided only if GLUE record for nameserver is needed.
tech	1..n	contactID	Technical contact	
authInfo	0..1	authInfo	Authorization information for a nsset.	If authorization information is empty string, it is generated by the server.
reportlevel	0..1	Nsset level	Level of technical checks performed on nsset. The higher the level is, the more detailed tests are executed. Zero means no technical test will be executed at all.	The level is set to zero by default if the reportlevel is not specified upon nsset creation.

Table 1.36. create keyset input parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	ID of keyset to be created	
dnskey	0..10	dnskeyT	DNS key record	
tech	1..n	contactID	Technical contact	

Element	Occurence	Valid value	Description	Note
authInfo	0..1	authInfo	Authorization information for a nsset.	If authorization information is empty string, it is generated by the server.

Table 1.37. DNS key record

Element	Occurence	Valid value	Description	Note
flags	1	unsignedShort	Key properties. supported values are 0, 256, 257.	See RFC 4034 for details
protocol	1	unsignedByte	Less than or equal 3	
alg	1	unsignedByte	Algorithm type	
pubKey	1	string	base64 encoded public key	

Expected return code is rc1000 or an error code in case of failure.

1.6. Delete

delete removes object from repository and has no output parameters.

Table 1.38. delete domain input parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Name of a domain to be deleted	

Table 1.39. delete contact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of contact to be deleted	

Table 1.40. delete nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of nsset to be deleted	

Table 1.41. delete keyset input parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	ID of keyset to be deleted	

Expected return code is rc1000 or an error code in case of failure.

1.7. Renew

renew extends validity of a domain object.

Table 1.42. renew domain input parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Name of domain to be renewed	
curExpDate	1	Date	Current expiration date of domain. Allthough the XML schemas permit a timezone appendix, the date must be given without this appendix (in the same form return by info domain command).	
period	0..1	period	Registration period for domain	Unit of time (year or month) is specified by attribute *unit*. If period is not specified, the default one is used.
valExDate	0..1	Date	Expiration date of validation of enum domain	This parameter is admissible only for enum domains.

Table 1.43. renew domain output parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Name of domain which was renewed	
exDate	0..1	Date	New expiration date after renewal	

Expected return code is rc1000 or an error code in case of failure.

1.8. Transfer

transfer changes ownership of an object. This command has no output parameters.

Table 1.44. transfer domain input parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Name of domain to be transferred	
authInfo	1	authInfo	Authorization information for a domain	Upon successfull transfer new authorization information is automatically generated.

Table 1.45. transfer contact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of a contact to be transferred	
authInfo	1	authInfo	Authorization information for a contact	Upon successfull transfer new authorization information is automatically generated.

Table 1.46. transfer nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of a nsset to be transferred	

Element	Occurence	Valid value	Description	Note
authInfo	1	authInfo	Authorization information for a nsset	Upon successfull transfer new authorization information is automatically generated.

Table 1.47. transfer keyset input parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	ID of a keyset to be transferred	
authInfo	1	authInfo	Authorization information for a nsset	Upon successfull transfer new authorization information is automatically generated.

Expected return code is rc1000 or an error code in case of failure.

1.9. Update

update changes properties of an object. This command has no output parameters.

Table 1.48. update domain input parameters

Element	Occurence	Valid value	Description	Note
name	1	domain	Name of domain to be updated	
admin	0..n	contactID	ID of admin contact to be added or deleted.	
nsset	0..1	change nssetID	Set nsset of domain	If the tag contains empty string, current nsset is taken away.
keyset	0..1	change keysetID	Set keyset of domain	If the tag contains empty string, current keyset is taken away.

Element	Occurence	Valid value	Description	Note
registrant	0..1	contactID	Set registrant	
authInfo	0..1	authInfo	Set authorization information	If authorization information is empty string, it is generated by the server.
valExDate	0..1	Date	Set expiration date of validation of ENUM domain	This can be set only for ENUM domain.
publish	0..1	boolean	Flag for publishing ENUM number and associated contact in public directory	This information is valid only for enum domains.

Table 1.49. update contact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of contact to be updated.	
name	0..1	postal line	New name	
org	0..1	optional postal line	Set organization	
street	0..3	optional postal line	Set street (up to three elements)	
city	0..1	postal line	Set city	
sp	0..1	optional postal line	Set state or province	
pc	0..1	postal code	Set postal code	
cc	0..1	country code	Set country code	
voice	0..1	telephone number	Set telephone number	
fax	0..1	telephone number	Set fax number	
email	0..1	emailCommaListType	Email address	
authInfo	0..1	authInfo	Authorization information for contact	If authorization information is empty

Element	Occurence	Valid value	Description	Note
				string, it is generated by the server.
disclose	0..n	disclose	The elements inside disclose element decide, which contact attributes are public or private.	
vat	0..1	VAT	Value added tax ID	
ident	0..1	ident	Unique identificator of contact	Type of identification is chosen by attribute *type*.
notifyEmail	0..1	emailUpdCommaListType	Set email address for notification emails	

Table 1.50. update nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of nsset to be updated.	
name	0..10	nameserver	FQDN of nameserver to be added or deleted	
addr	0..n	IP address	Address of nameserver, which is added.	Address should be provided only if GLUE record for nameserver is needed.
tech	0..n	contactID	Technical contact to be added or deleted	
authInfo	0..1	authInfo	Update authorization information for nsset	If authorization information is empty string, it is generated by the server.
reportlevel	0..1	Nsset level	Level of technical checks performed on nsset. The high-	The level is set to zero by default if the reportlevel is

Element	Occurence	Valid value	Description	Note
			er the level is, the more detailed tests are executed. Zero means no technical test will be executed at all.	not specified upon nsset creation.

Table 1.51. update keyset input parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	ID of keyset to be updated.	
dnskey	0..10	dnskeyT	DNS key records to be added or removed	
tech	0..n	contactID	Technical contact to be added or deleted	
authInfo	0..1	authInfo	Update authorization information for nsset	If authorization information is empty string, it is generated by the server.

Expected return code is rc1000 or an error code in case of failure.

1.10. sendAuthInfo

sendAuthInfo sends email containing authorization information to email address of appropriate contact. Appropriate contact is for object domain registrant, for objects nsset and keyset technical contact and for object contact it is the contact itself. The command has no output parameters.

Table 1.52. sendAuthInfo domain input parameters

Element	Occurence	Valid value	Description	Note
id	1	domain	Name of domain for which authorization information should be sent.	

Table 1.53. sendAuthInfo contact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of contact for which authorization information should be sent.	

Table 1.54. sendAuthInfo nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of nsset for which authorization information should be sent.	

Table 1.55. sendAuthInfo keyset input parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	ID of keyset for which authorization information should be sent.	

Expected return code is rc1000 or an error code in case of failure.

1.11. test

test requests technical check on nsset. This means that all nameservers in nsset will be tested in various aspects. Test suite is compossed from several tests which have different importances. The tests are done assynchronously in respect to the request. The client is informed about the results by poll message.

Table 1.56. test nsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of nsset which should be tested	
level	0..1	Nsset level	Set level of performed tests	If this is not set, the default level is value of reportlev-

Element	Occurence	Valid value	Description	Note
				el attribute of nsset.
name	0..n	domain	Domain FQDN to be tested with nsset	By default is tested nsset for all domains, which are bound to the nsset. By this directive we extent the default set of domains tested with nsset.

Expected return code is rc1000 or an error code in case of failure.

1.12. creditInfo

creditInfo returns registrar's credits for individual zones. This command is not related to any object and hasn't any input arguments.

Table 1.57. creditInfo output parameters

Element	Occurence	Valid value	Description	Note
zone	0..n	zone	Name of zone	
credit	0..n	credit	Remaining credit for the zone	

Expected return code is rc1000 or an error code in case of failure.

1.13. listDomains

listDomains does have neither input nor output parameters. It prepares a list of currently registered domains (only their FQDNs) by client. The list can be incrementally downloaded by *getResults* command.

Expected return code is rc1000 or an error code in case of failure.

1.14. listNssets

listNssets does have neither input nor output parameters. It prepares a list of currently registered nssets (only their IDs) by client. The list can be incrementally downloaded by *getResults* command.

Expected return code is rc1000 or an error code in case of failure.

1.15. listKeysets

listKeysets does have neither input nor output parameters. It prepares a list of currently registered keysets (only their IDs) by client. The list can be incrementally downloaded by *getResults* command.

Expected return code is rc1000 or an error code in case of failure.

1.16. listContacts

listContacts does have neither input nor output parameters. It prepares a list of currently registered contacts (only their IDs) by client. The list can be incrementally downloaded by *getResults* command.

Expected return code is rc1000 or an error code in case of failure.

1.17. domainsByNsset

domainsByNsset does not have output parameters. It prepares a list of domain FQDNs which are associated with nsset, identified by client provided nsset ID. The list can be incrementally downloaded by *getResults* command.

Table 1.58. domainsByNsset input parameters

Element	Occurence	Valid value	Description	Note
id	1	nssetID	ID of nsset.	

Expected return code is rc1000 or an error code in case of failure.

1.18. domainsByKeyset

domainsByKeyset does not have output parameters. It prepares a list of domain FQDNs which are associated with keyset, identified by client provided keyset ID. The list can be incrementally downloaded by *getResults* command.

Table 1.59. domainsByKeyset input parameters

Element	Occurence	Valid value	Description	Note
id	1	keysetID	ID of keyset.	

Expected return code is rc1000 or an error code in case of failure.

1.19. domainsByContact

domainsByContact does not have output parameters. It prepares a list of domain FQDNs which have admin, temporary or registrant contact, identified by client provided contact ID. The list can be incrementally downloaded by *getResults* command.

Table 1.60. domainsByContact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of contact.	

Expected return code is rc1000 or an error code in case of failure.

1.20. nssetsByContact

nssetsByContact does not have output parameters. It prepares a list of nsset IDs which have technical contact, identified by client provided contact ID. The list can be incrementally downloaded by *getResults* command.

Table 1.61. nssetsByContact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of contact.	

Expected return code is rc1000 or an error code in case of failure.

1.21. nssetsByNs

nssetsByNs does not have output parameters. It prepares a list of nsset IDs which have given nameserver. The list can be incrementally downloaded by *getResults* command.

Table 1.62. nssetsByNs input parameters

Element	Occurence	Valid value	Description	Note
id	1	nameserver	FQDN of nameserver.	

Expected return code is rc1000 or an error code in case of failure.

1.22. keysetsByContact

keysetsByContact does not have output parameters. It prepares a list of keyset IDs which have technical contact, identified by client provided contact ID. The list can be incrementally downloaded by *getResults* command.

Table 1.63. keysetsByContact input parameters

Element	Occurence	Valid value	Description	Note
id	1	contactID	ID of contact.	

Expected return code is rc1000 or an error code in case of failure.

1.23. getResults

getResults does not have input parameters. It returns a portion of list stored on server-side. The whole list can be downloaded by repetitive use of this command, until it returns no items. The list on server-side is initialized by any info function (listDomains, listContacts, listNssets, listKeysets, domainsByNsset, domainsByKeyset, domainsByContact, nssetsByContact, nssetsByNs, keysetsByContact).

Table 1.64. getResults output parameters

Element	Occurence	Valid value	Description	Note
item	0..n	List item	Item from list.	

Expected return code is rc1000 or an error code in case of failure.

1.24. Overview of used datatypes

Table 1.65. Summary of valid parameter types

Identifier	XML Schema data type	Other constraints
ROID	token, which must match pattern `(\w|_){1,80}-\w{1,8}.`	
count	unsignedLong	
zone	token of length from 1 to 255	
credit	decimal with maximum of 10 digits and from that 2 digits are used for fractional part.	
boolean	boolean	

Identifier	XML Schema data type	Other constraints
disclose	Enumeration of following elements: voice, fax, email, vat, ident, notifyEmail.	
clID	token, length from 3 to 16	
pw	token, length from 6 to 16	
lang	language	Must be one of the languages announced in greeting from server (currently only *cz* or *en*).
domain	token, length from 1 to 66	Domain name is case-insensitive, must not containt sequence of two or more dots, dot or "-" at the beginning. CZ domains must have suffix *.cz*. ENUM domains must match regular expression `([0-9]\.){1,9}0.2.4.e164\.arpa`.
domain status	domain status	token one of: ok, serverDeleteProhibited, serverRenewProhibited, serverTransferProhibited, serverUpdateProhibited, serverRegistrantChangeProhibited, serverBlocked, serverOutzoneManual, serverInzoneManual, expired, outzone, notValidated.
nsset status	nsset status	token one of: ok, linked, serverTransferProhibited, serverUpdateProhibited, serverDeleteProhibited.
keyset status	keyset status	token one of: ok, linked, serverTransferProhibited, serverUpdateProhibited, serverDeleteProhibited.
contact status	contact status	token one of: ok, linked, serverTransferProhibited, serverUpdateProhibited, serverDeleteProhibited.
nameserver	token, length from 4 to 255	FQDN is case-insensitive.

Identifier	XML Schema data type	Other constraints
contactID	token, length from 3 to 63	ID is case-insensitive. ID of contact must start with prefix `CID:`. Allowed characters are: letters, "_", "-", ":".
nssetID	token, length from 3 to 63	ID is case-insensitive. ID of nsset must start with prefix `NSSID:`.
change nssetID	token, length from 0 to 63	ID is case-insensitive. If ID has not zero length, ID of nsset must start with prefix `NSSID:`.
keysetID	token, length from 3 to 63	ID is case-insensitive. ID of keyset must start with prefix `KEYSID:`.
change keysetID	token, length from 0 to 63	ID is case-insensitive. If ID has not zero length, ID of keyset must start with prefix `KEYSID:`.
DNS key record	DNS key record,	DNS key record. Contains flags, protocol, alg and pubKey fields. See RFC 4034 for details.
msgID	token, length from 3 to 63	Token must be a number.
authInfo	normalizedString, length from 0 to 300	
period	unsignedShort from interval 1..99	The interval is further limited by server by upper bound of 10 years, counted from moment of command issue. And granularity of a value are years.
postal line	normalizedString, length from 1..255	
optional postal line	normalizedString, length from 0..255	
postal code	token, length from 0..16	
country code	token of length 2	Must be valid country code.
telephone number	token matching pattern `(\+[0-9]{1,3}\.[0-9]{1,14})?` and of length 0..17	

Identifier	XML Schema data type	Other constraints
email comma-separated list	emailCommaListType of maximal length 320	matching pattern `[^@, ]{1,64}@[^@, ]+(,[^@, ]{1,64}@[^@, ]+)*`
email comma-separated list	emailUpdCommaListType (optional emailCommaListType pattern) of maximal length 320	matching pattern `([^@, ]{1,64}@[^@, ]+(,[^@, ]{1,64}@[^@, ]+)*)?`
VAT	token, length from 0..20	
ident	token, length from 0..32	
IP address	token, length from 3..45	
Date and time	dateTime	
Date	date	
Nsset level	unsignedByte	Number from interval 0..10
Check message	normalizedString	
List item	token	
Poll message	The msg element according to formal definition can contain any XML mixed with text. However we are more strict, and we define precisely format of XML messages and mixed content type is not allowed. But these restrictions are not enforced by XML schemas.	See section about "poll request" command for definitions of possible poll messages.
Transaction ID	token of length from 3 to 64.	
Test name	token of length from 1 to 255.	
Note	string	

1.25. Return codes and messages

Table 1.66. Summary of valid parameter types

Code	Message	Reason
1000	Command completed successfully	Typical return code, when command is completed successfully.

Code	Message	Reason
1300	Command completed successfully; no messages	Answer to command "poll request", when there aren't any messages.
1301	Command completed successfully; ack to dequeue	Answer to command "poll request", when a message was returned. The message advises client to run "poll acknowledge" command, in order to delete message from queue.
1500	Command completed successfully; ending session	Response to successfull logout.
2000	Unknown command	Unknown command, not defined by schemas, was received. The whole XML is valid. Ussually it means that used object mapping is not known.
2001	Command syntax error	The XML is not valid.
2002	Command use error	Either logout is used when client is not logged in or login is used and client is already logged in.
2003	Required parameter missing	The XML is valid, but attribute required by context is missing.
2004	Parameter value range error	A numeric value is out of range, but the XML is still valid.
2005	Parameter value syntax error	A parameter does not match a pattern. For example contact ID must start with "CID:". These restrictions are not defined on level of schemas, but are explicitly checked by server. Therefore the XML is still valid, when this error occurs.
2100	Unimplemented protocol version	If in login client requests EPP protocol version other then 1.0.
2102	Unimplemented option	For example when transfer uses other option then "request" or unknown language is requested by client when logging in.

Code	Message	Reason
2200	Authentication error	Bad login name or password.
2201	Authorization error	Client is not authorized for certain operation, i.e. trying to modify object which is not owned by him.
2302	Object exists	An attempt to create object with ID, which is already present in database, was detected.
2303	Object does not exist	Operation on nonexistant object was requested.
2304	Object status prohibits operation	
2305	Object association prohibits operation	For example, contact associated with domain cannot be deleted.
2306	Parameter value policy error	
2400	Command failed	Opaque error.
2502	Session limit exceeded; server closing connection	Client openned to many connections to server.

Appendix A. XML Schema primitive and derived data types used in EPP protocol

normalizedString
: String which does not contain carriage return, line feed nor tab characters.

token
: It is a normalizedString with no leading or trailing spaces and with no internal sequences of two or more spaces.

language
: It is a token matching a pattern `[a-zA-Z]{1,8}(-[a-zA-Z0-9]{1,8})*` and defined in RFC 3066.

unsignedShort
: It is a non-negative integer with maximal value of 65535.

unsignedLong
: It is a non-negative integer with maximal value of 18446744073709551615.

decimal
: Set of numbers that can be obtained by multiplying an integer by a non-positive power of ten.

date
: It is characterized by template yyyy-mm-ddzzzzzz (y = year, m = month, d = day, z = timezone). The zone may be abbreviated to 'Z' in case of zero offset or may be left out altogether. Three examples of valid dates are: `2008-11-24`, `2008-09-05Z`, `2008-03-01+02:00`.

date and time
: It is characterized by template yyyy-mm-ddThh:mm:sszzzzzz (y = year, m = month, d = day, h = hour, m = minute, s = second, z = timezone). The zone may be abbreviated to 'Z' in case of zero offset or may be left out altogether.

unsignedByte
: It is a non-negative integer with maximal value of 255.

unsignedByte
: Either "true" or "false".

string
: Any sequence of characters.