Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

dot.swiss

.swiss/RegSys

.swiss Registration System (RegSys)

Edition 2: 18.01.2024
Entry into force: 24.04.2024

• 1. Version History

1.1 General Version History

Version Date	Comments
2013-08-20	initial version
2013-08-22	corrected URLs and addresses
2013-11-28	added launch phase information
2014-01-13	added formal launch phase specifications added XML Mark Data Extension information added a note on client SSL certificates for EPP connection clarified that no certification tests are to be passed to access the Production flavor some more minor changes
2014-01-20	added TLD-specific version history section
2014-02-17	changed IANA ID from optional to mandatory
2014-03-17	added the description of the EPP auction extension
2014-05-14	updated the *Creating New Registrar Accounts* section
2014-05-26	added note about potentially deviating Add Grace Period durations in specific launch phases
2014-07-03	added notes about "object mode" for variants and adjusted EPP IDN extension documentation
2014-09-24	improved documentation regarding the EPP IDN extension, documented new default language and new language update capabilities
2014-09-29	improved documentation regarding user access rights
2015-01-09	improved EPP examples, only show specific examples for the .swiss TLD
2015-01-12	added note about the e-mail capability of the issue system
2015-03-06	added information about domain review workflow and adjusted the application review workflow
2015-03-17	added information about available reports and how to download them
2015-03-18	added an overview of the control panel's functionality; split document to produce separate, dedicated registry and registrar documentation
2015-03-23	added sections describing the registry's EPP connection and Whois query policies
2015-04-01	added the description of the EPP fee extension
2015-07-21	added EPP policy prohibiting updates without changes
2015-07-24	added a section about promotions (only included in the document if promotions are

	enabled for the registry)
2015-08-03	added information about the contact eligibility extension (only included in the document if the contact eligibility extension is enabled for the registry)
2015-10-12	use global numbering of chapter/section headlines
2015-10-12	corrected example of EPP domain renew command with promotion code extension
2016-02-11	clarified information about auto-renewals of domains with "clientRenewProhibited" or "serverRenewProhibited" status values
2016-06-06	added information about the maximum number of DNSSEC data entries
2016-06-27	updated/enhanced documentation of fee extension implementation (only present if the registry supports it)
2016-08-12	added documentation about domain-based privacy (only present if the registry supports it)
2017-02-28	added information about the reserved contact handle prefix
2017-03-28	updated documentation of the fee extension implementation (only present if the registry supports it) for support of version urn:ietf:params:xml:ns:fee-0.15
2017-06-28	updated documentation of the fee extension implementation (only present if the registry supports it) for support of version urn:ietf:params:xml:ns:fee-0.21
2019-09-03	added documentation about the system's measures to comply with the EU's GDPR
2020-05-04	added documentation about the system's Two Factor Authentication capabilities
2020-07-23	updated documentation of the fee extension implementation (only present if the registry supports it) for added support of versions urn:ietf:params:xml:ns:epp:fee-1.0 (RFC 8748), urn:ietf:params:xml:ns:fee-0.23 and urn:ietf:params:xml:ns:fee-0.8, including explanations for the suitability of each version for the system's handling of premium domain pricing
2021-03-03	added documentation about the provisions regarding unhandled namespaces in EPP poll messages
2022-01-28	fixed domain policy documentation with regard to the optional billing contact
2022-04-07	changed documentation about the retention time for old, unreferenced contacts, which is now one year instead of three months
2022-07-12	updated promotion code EPP extension XML schema to allow promotion codes up to 64 characters long (instead of 20)
2023-01-20	added documentation about CDS/CDNSKEY support
2023-12-01	updated user permissions, adding Domain Manager and registry Accountant
2023-12-06	added information about the new contact eligibility extension version 1.1 (only included if the contact eligibility extension is enabled for the registry)

1.2 .swiss Version History

Version Date	Comments
2015-08-03	initial version

2015-08-07	added description of implementation of .swiss TLD policy (WS)
2015-08-12	integrated launch info (AAA)
2015-08-14	minor edits (WS)
2015-08-19	minor edits (WS), minor adjustments to news feed, publication
2021-04-09	added section about the Reporting Server

• 2. Guide for the Reader

The present document compiles some information needed for getting started with the CORE Registration System. It addresses both registry and registrars of the .swiss TLD and contains the following sections:

- *Overview of the CORE Registration System* briefly introduces the CORE Registration System and gives a short overview of its modules and features in order to create a general understanding of the system.
- *Technical Details of the CORE Registration System Instances* describes the various instances of the CORE Registration System hosted by CORE Internet Council of Registrars as well as needed access data. All technical staff interacting with the CORE Registration System should read this section in order to successfully connect to the system.
- *Users and Privileges* introduces the privilege concept of the CORE Registration System which allows to grant permissions to users working with the system. Registry employees who will create user accounts should be familiar with this concept.
- *The Control Panel* describes noteworthy aspects of the Control Panel which serves as a web interface to the CORE Registration System. It also explains the CORE RS Issue System which can, among others, be used for the communication between registry and registrars.
- *Reporting* contains information about the reporting capabilities of the CORE Registration System, covering both reports that can be created on demand in the Control Panel, as well as pre-generated reports which are available for (potentially automated) downloading.
- The sections *Registration Policies for the .swiss TLD* and following contain information relevant for registrars using the EPP interface of the CORE Registration System. These sections describe the general EPP compliance of the system, the policy configuration of the .swiss registry and the EPP extensions that are in place.
- Attached to this documentation are the relevant IDN tables for the .swiss TLD which are referred to in the *Registration Policies for the .swiss TLD* section. This attachment is intended as a reference for registrars.

Since CORE Internet Council of Registrars operates multiple TLDs with the CORE Registration System, those parts of this documentation that potentially differ for different TLDs are underlined or marked with a red bar. This is for convenience of the registrars having more than one of the CORE-operated TLDs in their portfolio.

• 3. Overview of the CORE Registration System

The CORE Registration System offers the full range of registry services for the successful operation of the .swiss TLD: Receipt of data from registrars concerning registration of domain names and name servers via EPP (SRS) or a web-based, interactive interface; dissemination of top-level domain (TLD) zone files (DNS); dissemination of contact or other information concerning domain name registrations (port-43 Whois, web-based Whois); DNS Security Extensions (DNSSEC). Additional benefits offered by the CORE Registration System are full support for Internet Protocol version 6 (IPv6), data escrow, registrar reports and support for Sunrise and Landrush phases. All of these are compliant with the new gTLD requirements.

The Shared Registry System (SRS) is the central coordinating instance in the overall system concept. It is the authoritative source of the domain, host and contact data, provides client/server-based access methods for the registrars and internal personnel to this data, is responsible for the zone generation, performs accounting and reporting, and feeds the Whois servers.

The SRS is responsible for managing the domain registrations by accepting requests for the creation, update and deletion of domains and related information from the registrars, who act on behalf of the registrants.

The CORE Registration System also supports launch phases like Sunrise and Landrush during which domain names can be applied for by trademark owners or other closed groups of registrants. The resulting applications can then be processed automatically, e.g. on a "first-come, first-served" basis, or manually by using the built-in CORE RS Issue System.

The CORE Registration System is fully compliant with established standards like Internet Engineering Task Force (IETF) Requests for Comments (RFCs) and can be customized for the specific needs of a top level domain, ensuring Internet Corporation for Assigned Names and Numbers (ICANN) gTLD standards compliance.

• 4. Technical Details of the CORE Registration System Instances

As registry backend operator, CORE Internet Council of Registrars hosts the CORE Registration System in its own datacenters in order to guarantee compliance with ICANN's stability and availability requirements. The various system instances of the CORE Registration System can be remotely accessed by employees of .swiss Registry, employees of .swiss registrars and external users (if desired by the registry).

This section describes the system instances ("flavors") provided by CORE Internet Council of Registrars as well as the information needed to access these instances.

• *4.1 System Flavors*

The CORE Registration System is delivered by CORE Internet Council of Registrars in the following instances, the so-called "flavors":

- Production flavor
- OT+E flavor
- Staging flavor

Each of these flavors is a standalone system including EPP interface, Whois service and Control Panel. There are no connections between the flavors, i.e., operations performed with one flavor will not have any side-effects on the other flavors.

In the following, the purpose of the provided flavors is described in detail.

• *4.1.1 Production*

The Production flavor is used for the productive operation of the .swiss registry.

.swiss Registry, CORE Internet Council of Registrars (as registry backend operator) and .swiss registrars will get access data for this flavor. Objects created with this CORE Registration System instance will appear in the zone file of the .swiss TLD and will thus be charged.

No certification tests are to be passed in order to access the Production flavor.

• *4.1.2 OT+E (Operational Test and Evaluation)*

The OT+E flavor is a sandbox environment used for registrar tests. In particular, .swiss registrars may use the OT+E flavor in order to test the EPP interface of the CORE Registration System and to adapt their client software to this interface without interfering with the actual production environment.

The OT+E flavor and the Production flavor of the CORE Registration System will run the same software revision in order to guarantee a realistic sandbox environment. However, the configurations of the OT+E flavor and the Production flavor may be different in order to provide a convenient test environment; for example, periods and deadlines used by the Production flavor (such as the Add Grace Period) are significantly shortened on the OT+E flavor. For details, please refer to the *Registration Policies* section.

Besides .swiss Registry, CORE Internet Council of Registrars (as registry backend operator) and .swiss registrars, also external actors may get access data for this flavor if desired by the registry. The usage of the OT+E flavor is free of charge for all actors using this flavor.

• 4.1.3 Staging

The Staging flavor is used as an acceptance system for .swiss registrars: When a new software revision that contains substantial changes or new features has been accepted by the registry, this revision is installed on the Staging flavor in order to provide an acceptance environment for .swiss registrars. Once the new software revision has been thoroughly tested by the registrars, it will be installed on the OT+E flavor and on the Production flavor.

.swiss Registry, CORE Internet Council of Registrars (as registry backend operator) and .swiss registrars will get access data for this flavor. The usage of the Staging flavor is free of charge for all actors using this flavor.

• 4.2 Access Data

Each flavor of the CORE Registration System provides three services that may be remotely accessed: The Control Panel, the EPP interface and the Whois service. This subsection explains how to access these services. For each service, two URLs or addresses are given: an official one which is maintained by the .swiss registry (.swiss Registry) and an alternative one which is maintained by the registry backend operator (CORE Internet Council of Registrars).

• 4.2.1 The Control Panel

The Control Panel can be accessed using any web browser via the following URLs:

- Production flavor:
  - Official URL: https://cp.nic.swiss
  - Alternative URL: https://cp.swiss.rs.corenic.net
- OT+E flavor:
  - Official URL: https://cp-ote.nic.swiss
  - Alternative URL: https://cp.ote.swiss.rs.corenic.net
- Staging flavor:
  - Official URL: https://cp-staging.nic.swiss
  - Alternative URL: https://cp.staging.swiss.rs.corenic.net

The credentials needed to log on to the Control Panel are communicated over a separate channel.

• 4.2.2 The EPP Interface

The EPP interface can be accessed using an EPP client via the following addresses:

- Production flavor:
  - Official address: epp.nic.swiss
  - Alternative address: epp.swiss.rs.corenic.net
  - Port: 700
- OT+E flavor:

    - Official address: epp-ote.nic.swiss
    - Alternative address: epp.ote.swiss.rs.corenic.net
    - Port: 700
- Staging flavor:
    - Official address: epp-staging.nic.swiss
    - Alternative address: epp.staging.swiss.rs.corenic.net
    - Port: 700

The credentials needed to use the EPP interface are generated when a new registrar account is created in the CORE Registration System.

Client SSL certificates are optional in the EPP implementation of the CORE Registration System, i.e., a certificate may be used for connection, but it is not mandatory. However, if a certificate is provided, it must be issued by a widely trusted CA, e.g., Verisign or Thawte in order to be trusted by the SRS.

4.2.3 The Whois Service

The Whois service can be accessed using either a command line tool or a web browser. When accessing the Whois service via the command line, the following addresses must be used:

- Production flavor:
    - Official address: whois.nic.swiss
    - Alternative address: whois.swiss.rs.corenic.net
    - Port: 43
- OT+E flavor:
    - Official address: whois-ote.nic.swiss
    - Alternative address: whois.ote.swiss.rs.corenic.net
    - Port: 43
- Staging flavor:
    - Official address: whois-staging.nic.swiss
    - Alternative address: whois.staging.swiss.rs.corenic.net
    - Port: 43

As an alternative, the Whois service can be accessed via a web interface which also offers extended search facilities. This web interface can be reached using the following URLs:

- Production flavor:
    - Official URL: https://whois.nic.swiss
    - Alternative URL: https://whois.swiss.rs.corenic.net
- OT+E flavor:
    - Official URL: https://whois-ote.nic.swiss
    - Alternative URL: https://whois.ote.swiss.rs.corenic.net
- Staging flavor:
    - Official URL: https://whois-staging.nic.swiss
    - Alternative URL: https://whois.staging.swiss.rs.corenic.net

5. Users and Privileges

There are different kinds of actors that may use or interact with the CORE Registration System. These actors can be (human) persons or (technical) systems and may belong to the registry (.swiss Registry), the registry backend operator (CORE Internet Council of Registrars), a .swiss registrar or an external organization.

Depending on the tasks an actor, in the following referred to as "user", is supposed to perform, he needs appropriate access rights which are represented as privileges in the CORE Registration System. In the following, these privileges and the associated permissions are explained.

• 5.1 Introduction to Privileges

A privilege represents a role, i.e. a set of permissions, regarding a particular service. For example, a user of the CORE Registration System may have the permission to perform administrative tasks regarding the Control Panel. Another user may have the permission to access the Whois service in order to perform the tasks of a law enforcement officer. A user may have multiple privileges associated, making it possible to act in different roles.

The roles that may be assigned to users are explained in detail in the next section.

• 5.2 Valid Privileges

The following table gives an overview of the privileges that may be granted with the CORE Registration System. The columns of the table represent the available services, whereas the rows represent the roles that may be assigned to users.

Each user has an associated user type which is one of the following:

- Registry: Users working on behalf of .swiss Registry
- Registrar: Users working on behalf of a .swiss registrar
- External: External users working neither for .swiss Registry nor for a .swiss registrar

Each cell in the following table represents the privilege defined by the service of the table column and the role of the table row. The cell contains the user type(s) for which this privilege may be granted.

	Control Panel	EPP interface	Whois service
Administrator	Registry	Registry	
Assistant	Registry		
Manager	Registrar		
Accountant	Registry, Registrar		
Domain Manager	Registry, Registrar		
User	Registry, Registrar	Registrar	
Law Enforcement			External
Security			External
Whitelisted			Registry, Registrar, External

For example: The privilege (Control Panel, User) may be granted to Registry users and to Registrar users.

Note that users that have the Administrator role with respect to a particular service may perform any task regarding this service; i.e., the Administrator role always subsumes all other roles being valid for the respective service.

Each user (human or system) represented in the CORE Registration System must have assigned at least one privilege; empty privilege sets are not allowed.

• 5.3 Semantics of Privileges

Each privilege represents a certain set of permissions the user has with respect to the privilege's service. These permissions are now explained briefly.

• 5.3.1 Privileges Regarding the Control Panel Service

• 5.3.1.1 User Types

• 5.3.1.1.1 Registry Users

Administrator users may perform tasks that are crucial for the fundamental operation of the registry like, e.g., the definition of launch phases or the configuration of blocked or reserved domain names. The Administrator role subsumes the Assistant, Domain Manager, Accountant, and {rsRoleUser roles.

Assistant users may perform everyday tasks requiring a certain responsibility like, e.g., creation of registrar accounts or creation of domains on behalf of a registrar. The Assistant role subsumes the Domain Manager and User roles.

Domain Manager users may manage all registry objects (i.e., domains, applications, contacts, and hosts) independent of the sponsoring registrar. They have full write access on these objects. Domain Manager users are also able to impersonate registrar users, with the exception that registrar Accountant users can only be impersonated, if the registry user also has the Accountant role. The Domain Manager role subsumes the User role.

Accountant users get access to accounting-related items like, e.g., tariff management, invoices, and booking entries. The Accountant role subsumes the User role.

Users with the User role may perform everyday tasks requiring less responsibility . They are not able to alter or create any registry objects (i.e., domains, applications, contacts, and hosts) although they are able to view these objects.

• 5.3.1.1.2 Registrar Users

Manager users may create other user accounts. The Manager role subsumes the Domain Manager and User roles.

Domain Manager users may manage all of the registrar's objects (i.e., domains, applications, contacts, and hosts). They have full write access on these objects. The Domain Manager role subsumes the User role.

Accountant users get access to accounting-related items like, e.g., invoices and booking entries. The Accountant role subsumes the User role.

Users with the User role may perform everyday tasks like, e.g., creation of domains or working on issue system tickets.

• 5.3.2 Privileges Regarding the EPP Interface

Systems with the (EPP interface, User) privilege may connect via the EPP interface and perform all EPP operations. Note that this privilege is automatically assigned to all systems that try to connect to the EPP interface and are whitelisted for this service.

Systems with the (EPP interface, Administrator) privilege may connect via the EPP interface and perform all EPP operations on behalf of other registrars. Note that such systems can never perform any EPP operation for themselves, because they belong to the registry which is not able to perform EPP operations.

• 5.3.3 Privileges Regarding the Whois Service

The Law Enforcement role is assigned to law enforcement officers who shall be granted access to the searchable Whois service.

The Security role is assigned to people working for security organizations like CERTs and CSIRTs and who shall be granted access to the searchable Whois service.

The Whitelisted role is assigned to systems for which the rate limiting of the Whois service shall not apply. Note that this privilege is automatically assigned to all systems that try to connect the Whois service and are whitelisted for this service.

- ### *5.3.4 Access Rights*
- #### *5.3.4.1 Control Panel*

Registry Users:

Operation	Administrator	Assistant	Domain Manager	Accounta
domain/application				
- search/view	✓	✓	✓	✓
- create	✓	✓	✓	-
- clone	✓	✓	✓	-
- edit	✓	✓	✓	-
- delete	✓	✓	✓	-
- check (domain)	✓	✓	✓	✓
- transfer (domain)	✓	✓	✓	-
- bulk transfer (domain)	✓	✓	✓	-
- renew (domain)	✓	✓	✓	-
contact				
- search/view	✓	✓	✓	✓
- create	✓	✓	✓	-
- clone	✓	✓	✓	-
- edit	✓	✓	✓	-
- delete	✓	✓	✓	-
host				
- search/view	✓	✓	✓	-
- create	✓	✓	✓	-
- edit	✓	✓	✓	-
- delete	✓	✓	✓	-
registrar				
- search/view	✓	✓	✓	✓
- create	✓	✓	-	-
- clone	✓	✓	-	-
- edit	✓[1]	✓[1]	-	-

- delete	✓	✓	-	-
- impersonate	✓	✓	✓	-
user				
- search/view	✓	✓	✓	✓
- create	✓	✓	-	-
- edit	✓	✓	-	-
- delete	✓	✓	-	-
- impersonate	✓	✓	✓	-
- password reset	✓	✓	✓	✓
- reset TOPT	✓	✓	-	-
- onboarding	✓	✓	-	-
accounting				
- invoices	✓	-	-	✓
- book entry search	✓	-	-	✓
- deposit	✓	-	-	✓
- tariff view	✓	✓	✓	✓
- tariff management	✓	-	-	✓
report				
- ICANN	✓	✓	-	-
issue				
- dashboard	✓	✓	✓	✓
- search/view	✓	✓	✓	✓
- create/management	✓	✓	✓[2]	✓[2]
settings				
- my profile	✓	✓	✓	✓
- registrar profile (incl. EPP and Whois settings)	-[1]	-[1]	-[1]	-[1]
- rules and variables view	✓	✓	-	-
- rules and variables edit	✓	✓	-	-
- launch phases	✓	✓	✓	✓

- launch phases management	✓	-	-	-

Registrar Users

Operation	Manager	Domain Manager	Accountant	User
domain/application				
- search/view	✓	✓	✓	✓
- create	✓	✓	-	-
- clone	✓	✓	-	-
- edit	✓	✓	-	-
- delete	✓	✓	-	-
- check (domain)	✓	✓	✓	✓
- transfer (domain)	✓	✓	-	-
- bulk transfer (domain)	✓	✓	-	-
- renew (domain)	✓	✓	-	-
contact				
- search/view	✓	✓	✓	✓
- create	✓	✓	-	-
- clone	✓	✓	-	-
- edit	✓	✓	-	-
- delete	✓	✓	-	-
host				
- search/view	✓	✓	✓	✓
- create	✓	✓	-	-
- edit	✓	✓	-	-
- delete	✓	✓	-	-
registrar				
- search/view	-	-	-	-
- create	-	-	-	-
- clone	-	-	-	-
- edit	-[1]	-	-	-
- delete	-	-	-	-

- impersonate	-	-	-	-
user				
- search/view	✓	-	-	-
- create	✓	-	-	-
- edit	✓	-	-	-
- delete	✓	-	-	-
- impersonate	-	-	-	-
- password reset	✓	-	-	-
- reset TOPT	✓	-	-	-
- onboarding	-	-	-	-
accounting				
- invoices	-	-	✓	-
- book entry search	-	-	✓	-
- deposit	-	-	-	-
- tariff view	-	-	✓	-
- tariff management	-	-	-	-
report				
- ICANN	-	-	-	-
issue				
- dashboard	✓	✓	✓	✓
- search/view	✓	✓	✓	✓
- create/management	✓[2]	✓[2]	✓[2]	✓[2]
settings				
- my profile	✓	✓	✓	✓
- registrar profile (incl. EPP and Whois settings)	✓[1]	-	-	-
- rules and variables view	-	-	-	-
- rules and variables edit	-	-	-	-
- launch phases	✓	✓	✓	✓
- launch phases management	-	-	-	-

Notes:

1. The edit registrar (registry) functionality is different from registrar profile (registrar) as the

former allows access to the following fields (i.e., changes to these fields cannot be made by the registrar itself but must be requested via the registry's support channels):

  - Name
  - Registrar ID
  - IANA ID
  - Minimum Account Balance
  - all VAT-related data
  - EPP Login Name
  - Number of Concurrent EPP Connections

2. For details of the issue management and what type of issues can be managed by which user see the separate section "Ticket Details."

• *5.3.4.2 EPP Interface*

The EPP interface does not support different access rights for user types. A registrar EPP user can execute arbitrary EPP commands. Administrative EPP commands are not supported, neither for registrars nor for registry personnel.

• *5.3.4.3 Whois Service*

Next to the publicly available Whois services, a searchable Whois exists for which a login with credentials is necessary. Only users with the privilege "Law Enforcement" or "Security" are allowed to log in to the Whois and use the extended search functionality. Credentials for this functionality are issued by the registry upon request, given that sufficient proof of eligibility is provided.

• 6. The Control Panel

• *6.1 Overview*

The Control Panel has been designed to be largely self-explanatory and usable intuitively. The menu bar at the top of the page may be used to navigate to the various sections which correspond to typical registrar tasks:

- The "Domains" section offers functions to search, check, create, modify, delete and restore domain or application objects (for applications, a limited set of operations is available). In order to work on an already existing domain or application, use the "search" function to find the domain/application first, then use the context menu supplied in the search results to select the desired operation on the respective object. The "Domains" section also offers means for transferring domains, and to check a domain's transfer status.
- Likewise, the "Contacts" and "Hosts" sections allow the creation, modification and deletion of contact and host objects, respectively.
- The "Users" section offers functions to create, modify and delete Control Panel users. In order to work on an already existing user, use the "search" function to find the user first, then use the context menu supplied in the search results to select the desired operation on the respective user.
- The "Accounting" section offers access to accounting related information, such as
  - Invoices (note that some registries may choose to use external invoicing systems, in which case this section will not show any invoices)
  - Book Entries (each billable operation in the system creates a book entry, which will show up in this section; book entries can be searched for by various criteria)
  - Tariffs (offers an overview of all applicable tariffs in the system, distinguished by launch phase and other criteria).
- The "Reports" section offers download links to the pre-generated reports.

- The "Issues" section offers access to the system's built-in Issue (ticketing) System. See the section "The CORE RS Issue System" for details.
- The "Settings" section offers access to some global settings, such as
    - Editing the current user's profile,
    - Editing the current registrar's profile.
    - Viewing details regarding the system's configured launch phases, including those phases that have not started yet.

Additionally, some useful information is always present in the header section of the Control Panel, including

- the name of the logged in user and his level,
- the name of the user's registrar (if he/she belongs to a registrar) and
- the registrar's current account balance.

• 6.2 Two Factor Authentication (2FA)

The Control Panel optionally allows to secure login via a second factor: the time based one time password (TOTP) as defined in RFC 6238. The setting can be changed on a per-registrar basis. Whenever a user with 2FA activated but not yet initialized logs into the Control Panel a page to set-up the second factor is displayed. This features a QRCode to automatically initialize the TOTP as well as all required data to manually configure it. Many mobile phone apps or browser extensions support the RFC 6238 one-time passwords, e.g., the Google Authenticator or FreeOTP. To verify the successful configuration an TOTP needs to be supplied to proceed and finalize the log-in.

In case the second factor gets lost for whatever reason, the registry support needs to be contacted to reset the TOTP to its initial state. For security reasons there is no automatic recovery as exists for the password.

The .swiss Registry does currently not support 2FA.

• 6.3 The CORE RS Issue System

This section shortly describes the purpose and the basic concepts of the issue system being integrated into the CORE Registration System, the CORE RS Issue System for short. In general, issue systems (also called issue tracking systems or (trouble) ticket systems) maintain small work packages (issues), reflect their current state as well as their history and allow to assign them to employees of an organization in order to facilitate the structured handling of these work packages.

The CORE RS Issue System has three main purposes:

1. Facilitate the communication between registry and registrars.
2. Support the manual review of domain applications in the scope of launch phases.
3. Allow a convenient handling of abuse reports.

In the following, the terminology used in the CORE RS Issue System is described.

• 6.3.1 Terminology

The terminology of the CORE RS Issue System is based on the terminology of other widespread and well-known issue systems. In the following, this terminology is shortly recapitulated.

• 6.3.1.1 Ticket

A ticket is a basic unit of the CORE RS Issue System. It reflects a small work package like, e.g., the manual review of a domain application or the handling of an abuse report. Each ticket may contain various information about the work package it represents such as:

- the *type*, i.e. the category of the ticket
- the *priority* of the work package
- a short *summary* of the work package

- a detailed *description* of the work package
- the *reporter* who created the ticket
- the *due date* of the work package
- the *audience*, i.e., the employees or third parties who may observe the ticket state
- *attachments* needed to handle the ticket
- *comments* added by people who contribute to the ticket

6.3.1.2 Workflow State

At each point of time in its lifecycle, a ticket assumes one of the following workflow states:

- *Open:* Tickets that are currently not in progress, i.e., not assigned to an individual person. Most of the times this will be the case directly after ticket creation, but other workflows are also possible.
- *In Progress:* Tickets that have been assigned to individuals who actually work on this ticket.
- *Resolved:* Tickets that have been completed from the view of their assignee.
- *Closed:* Tickets that have been completed from the view of their reporter.

The current workflow state is assigned by the issue system after having been triggered by the user working on the respective ticket. For instance, when a user "takes" a ticket via the Control Panel, it is automatically put in progress.

6.3.1.3 Resolution

When a ticket is resolved, the outcome of the ticket processing is assigned to the ticket in form of a resolution label or resolution for short. The resolutions supported by the CORE RS Issue System are described in the next subsection.

6.3.2 Ticket Details

6.3.2.1 Types

In the following, for each ticket type of the CORE RS Issue System it is described, who is allowed to view and who is allowed to edit tickets of this type.

- *Application:* Tickets of this type are automatically created by the CORE RS Issue System when domain applications are received by the system in the scope of a launch phase in which the manual review of applications is necessary. This manual review process is the work package being captured by the ticket.
  - *Ticket may be viewed by:* registry and applying registrar
  - *Ticket may be edited by:* registry
- *Domain Review:* Tickets of this type are automatically created by the CORE RS Issue System when domain applications are received by the system with a questionable domain name. The manual review process of this domain name is the work package being captured by the ticket.
  - *Ticket may be viewed by:* registry and applying registrar
  - *Ticket may be edited by:* registry
- *General:* This is a general purpose ticket type for the communication between registry and registrar or between registry and registry backend operator.
  - *Ticket may be viewed by:* registry and affected registrar
  - *Ticket may be edited by:* registry and affected registrar
- *Contention:* If multiple applications for the same domain name exist at the end of a launch phase, a contention ticket is created by the issue system for each of these applications. A registry employee must then manually select exactly one of these tickets and resolve it with the resolution accepted which causes the CORE Registration System to accept the corresponding domain application and to convert it into a domain. All other contention tickets of the same contention set are then automatically resolved as rejected and the corresponding domains

applications are rejected as well.

  - *Ticket may be viewed by:* registry
  - *Ticket may be edited by:* registry

- *Abuse Report:* Abuse of domain names can be reported using a web formular. Such an abuse report generates an abuse report ticket in the CORE RS Issue System.
  - *Ticket may be viewed by:* registry and affected registrar
  - *Ticket may be edited by:* registry and affected registrar

6.3.2.2 Assignment and Visibility

A ticket may be assigned to a user or to a group:

- *User Tickets:* A ticket may be assigned to a user to indicate that this user is working on the ticket.
- *Group Tickets:* For the registry as well as for each registrar using the CORE Registration System, a dedicated group exists in the CORE RS Issue System. This concept allows for assigning a ticket to a group instead of to a particular user. The rationale behind this is to hide the internal structures of the registry or of a registrar: A registrar user should not have to know which particular user on the registry side is responsible for handling his request and vice versa. Consequently, if a registrar user wants to assign a ticket to a registry user, he assigns it to the registry group instead and each registry user may then "take" this ticket and handle it.

Moreover, note the following details regarding the assignment and the visibility of CORE RS Issue System tickets:

- If a ticket is assigned to a specific user, this user must be in the same group as the assigner.
- Group tickets are displayed on the dashboard for each user belonging to the group. However, tickets to be processed by first-level registry staff are separated from tickets to be processed by second-level registry staff.
- Group tickets may not be put in progress. To do so, the ticket must first be assigned to a specific user.
- A ticket that has been visible to a specific registrar user is visible to all users of this registrar and only to these. The reason is that the issue system should serve for communication between registrar and registry and not for communication between registrars.
- A registry user may remove a ticket from the visibility of its current registrar group and then assign it to another registrar group (this includes unassigning the ticket from this registrar group or user). Prior to the removal, a warning is displayed to make the registry user aware of potential data leaks.

6.3.2.3 Resolutions

The following table shows which resolutions can be chosen for which ticket type:

Resolution	Application	Domain Review	General	Contention	Abuse Report
None	x	x	x	x	x
Tentatively Valid	x	x			
Tentatively Invalid	x	x			
Valid	x	x			
Invalid	x	x			
Missing Information	x	x			
Undecided	x	x			

Withdrawn	x	x			
Rejected			x	x	x
Accepted				x	
Duplicate			x		x
Incomplete			x		x
Solved			x		x

The resolution labels have the following semantics:

- *None:* No specific resolution is set.
- *Tentatively Valid:* The application or domain under review is tentatively labeled as valid by a user of the registry. In other words, the user suggests to label it as valid.
- *Tentatively Invalid:* The application or domain under review is tentatively labeled as invalid by a user of the registry. In other words, the user suggests to label it as invalid.
- *Valid:* The application or domain under review is labeled as valid.
- *Invalid:* The application or domain under review is labeled as invalid.
- *Missing Information:* The application or domain under review cannot be labeled as valid or invalid because there is information missing to take this decision.
- *Undecided:* The application or domain under review is neither labeled as valid nor as invalid.
- *Withdrawn:* The application or domain under review has been withdrawn and thus needs not to be labeled as valid or invalid.
- *Rejected:* This resolution indicates that the work package defined by the respective ticket will not be completed. When rejected is chosen as a resolution, a comment should be added to the ticket that describes the reason for the rejection.
- *Accepted:* This resolution can only be chosen for contention tickets and signifies that the application being linked to this ticket shall be accepted as a domain.
- *Duplicate:* The work package captured by this ticket has already been captured by another ticket.
- *Incomplete:* The description of the ticket is not sufficient to complete the work package being captured by this ticket.
- *Solved:* The work package of the ticket has been completed.

7. Reporting

7.1 Overview

The TANGO system offers a number of reporting features that allow both registry staff and registrars to obtain detailed information about their object inventory, ongoing registry processes, and accounting information.

Within the Control Panel, various provided search pages support the interactive search for objects by various criteria:

- "Domains → Domain Search" allows searching for active and/or deleted domains. The results may be exported as a CSV file.
- "Domains → Applications Search" allows searching for domain applications.
- "Domains → Transfer Summary" allows searching for pending and completed domain transfers, both incoming and outgoing.

- "Contacts → Search" allows searching for active and/or deleted contact objects.
- "Hosts → Search" allows searching for active and/or deleted host objects.
- "Accounting → Book Entry Search" allows searching for accounting book entries (available to users with the Accountant role). The results may be exported as an XML or CSV file.
- "Accounting → Invoices" allows searching for invoices and downloading them (available to users with the Accountant role).

In addition, the system offers pre-generated reports for download in the "Reports" section of the Control Panel.

- *Daily object inventory reports*, consisting of a ZIP archive containing several CSV files with **all data of all objects sponsored by the registrar at the report's reference date**.
- *Daily auto-renewal reports*, consisting of a CSV file listing all domains that were **auto-renewed during the 24 hours before the report's reference date**; more precisely, these domains left their auto-renew grace period during these 24 hours, causing an auto-renewal to be charged to the registrar's account.
- *Daily expiration reports*, consisting of a CSV file listing all domains that will **reach their expiration date within two months after the report's reference date**.

7.2 Downloading pre-generated reports

Interactively, the pre-generated reports described above may be downloaded by clicking the respective links displayed on the reports page in the aforementioned "Reports" section of the Control Panel. The system automatically utilizes the logged in user's authorization to find and grant access to the available reports.

However, some registrars may want to download these reports in a non-interactive fashion, e.g. for a regular object inventory synchronization, or to detect daily auto-renewal charges which may then trigger charges to the respective registrants. For this purpose, these reports are also available for automatic download from specific URLs, according to the following patterns:

- Daily object inventory reports are downloadable from **https://rs.nic.swiss/download/registrar/*<registrar-id>*/inventory/*<YYYY-MM-DD>*** where *<registrar-id>* is the registrar's ID in TANGO (not the IANA ID), and *<YYYY-MM-DD>* is the report's reference date.
- Daily auto-renewal reports are downloadable from **https://rs.nic.swiss/download/registrar/*<registrar-id>*/autorenew/*<YYYY-MM-DD>*** where *<registrar-id>* is the registrar's ID in TANGO (not the IANA ID), and *<YYYY-MM-DD>* is the report's reference date.
- Daily expiration reports are downloadable from **https://rs.nic.swiss/download/registrar/*<registrar-id>*/expiration/*<YYYY-MM-DD>*** where *<registrar-id>* is the registrar's ID in TANGO (not the IANA ID), and *<YYYY-MM-DD>* is the report's reference date.

Note that the registrar ID can be determined by navigating to "Settings" → "Registrar Profile" and inspecting the "Registrar Data" section at the top of the page.

So, for example, the daily object inventory report of registrar "reg-12345" for June 10th, 2014 may be downloaded via

https://rs.nic.swiss/download/registrar/reg-12345/inventory/2014-06-10

In the case of auto-renew and expiration reports, the downloaded file will have the MIME type **"text/csv"**. For the object inventory report, the downloaded file will have the MIME type **"application/x-zip-compressed"**.

7.3 Authentication

The reports may only be downloaded by authorized parties; in particular, pre-registrar reports may of course only be downloaded by the registrar to which the reports belong. Since no Control Panel login context is available for these direct downloads, requests to the URLs above must include authentication information, using the "Authorization" HTTP header, as defined by the **HTTP basic**

authentication mechanism (RFC 2617). The credentials to be used are the same as for the interactive login of any of the registrar's users at the respective production Control Panel. This means that, if a registrar user's user name and password for the Control Panel login are known, these same credentials may be used in the "Authorization" HTTP basic authentication header to download that registrar's reports from the above URLs.

7.4 Report Retention

Registrar reports will only be retained for a limited amount of time. Usually, they will be deleted (and thus no longer be available for download) about **seven days** after their reference date.

• 8. Registration Policies for the .swiss TLD

• *8.1 General EPP Compliance*

The CORE Registration System implements EPP (the Extensible Provisioning Protocol) in full compliance with the following RFCs:

- RFC 5730 - Extensible Provisioning Protocol (EPP), http://tools.ietf.org/html/rfc5730
- RFC 5731 - Extensible Provisioning Protocol (EPP) Domain Name Mapping, http://tools.ietf.org/html/rfc5731
- RFC 5732 - Extensible Provisioning Protocol (EPP) Host Mapping, http://tools.ietf.org/html/rfc5732
- RFC 5733 - Extensible Provisioning Protocol (EPP) Contact Mapping, http://tools.ietf.org/html/rfc5733
- RFC 5734 - Extensible Provisioning Protocol (EPP) Transport over TCP, http://tools.ietf.org/html/rfc5734
- RFC 3915 - Domain Registry Grace Period Mapping for the Extensible Provisioning Protocol (EPP), http://tools.ietf.org/html/rfc3915
- RFC 5910 - Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP), http://tools.ietf.org/html/rfc5910

The concrete behavior of an instance of the CORE Registration System is governed by some (partially configurable) policy settings, which are described below.

Moreover, for the provisioning and management of domain name registrations and applications during the launch of the .swiss registry (or for specific modes of registration), the following RFC is supported:

- RFC 8334 - Launch Phase Mapping for the Extensible Provisioning Protocol (EPP), http://tools.ietf.org/html/rfc8334.txt

• *8.2 EPP Connection Policies*

Access to the registry system's EPP interface is restricted as follows:

- A registrar's EPP client may only connect to the EPP interface from one of the registrar's **whitelisted IP addresses** or address ranges. The EPP IP whitelist addresses and address ranges can be managed by the registrar via the Control Panel under "Settings" → "Registrar Profile". Please note that there is a separate IP whitelist for unlimited Whois queries (see "Whois Query Policies" below).
- A registrar's EPP client must authenticate itself using the **EPP login name** assigned by the registry during the onboarding process, and **EPP password** set by the registrar. The EPP password can be set via the Control Panel under "Settings" → "Registrar Profile".
- Use of an SSL client certificate by a registrar's EPP client is not mandatory. However, *if* a client certificate is used, it must be issued by a widely accepted Certificate Authority. The common

name of the certificate is not checked by the EPP server.

- A registrar's **concurrent number of open EPP connections** is limited; the maximum number of parallel connections is set by the registry. The currently configured number for a registrar can be viewed in the Control Panel under "Settings" → "Registrar Profile".
- The EPP commands sent over a single EPP connection are **rate limited**; the system is configured to allow up to **500 EPP commands per minute**.

8.3 Whois Query Policies

The registry's Whois service on port 43 is rate limited for the general public. However, registrars may denominate certain IP address ranges from which they will have unlimited access to the port 43 interface.

The Whois IP whitelist addresses and address ranges can be managed by the registrar via the Control Panel under "Settings" → "Registrar Profile". Please note that there is a separate IP whitelist for access to the EPP interface (see "EPP Connection Policies" above).

8.4 Registration Policies

The following policy settings are currently implemented for domain registrations in the .swiss TLD production shared registration system (SRS):

8.4.1 General Policies

- Update commands not changing any data will be prohibited.

8.4.2 Host / Name Server Policies

- The system uses host objects (EPP: "hostObj"). Host attributes (EPP: "hostAttr") are not supported.
- A host name must have at least two labels. The overall length of the host name must not exceed 255 characters (neither in U-, nor in A-label representation).
- Host renaming is supported.
- Host objects cannot be deleted as long as they're still used as name servers in any domain (even domains sponsored by other registrars).
- A host may have up to 12 IP addresses (IPv4 and IPv6 combined).
- In-Zone host objects (with names ending in .swiss) must have at least one IP address.
- IPs must be unique within a host object, i.e. no IP may occur more than once in a single host object. However, the same IP address may occur in multiple host objects.
- No verification of the correct delegation/zone configuration of the name servers used in a domain is performed. In particular, a domain's name servers do not have to be pre-configured before the domain can be registered.

8.4.3 Contact Policies

- The system supports both internationalized (ASCII-only) postal data and localized (full Unicode character set) postal data. One set of postal data is mandatory, which may be the internationalized or the localized version. It is also allowed to specify both.
- Contact transfers are not supported.
- The handle prefix "reg" (case insensitive) is reserved for registry use, e.g. for automatically assigned contact handles when contacts are created in the Control Panel. Registrars may therefore not create contacts using handles starting with this prefix.

8.4.4 Domain Policies / Grace Periods

8.4.4.1 General Domain Policies

- Domain names must have exactly two labels, the rightmost of which must be "swiss".

- A domain must have exactly one registrant, one administrative contact, and one technical contact. The billing contact is optional.
- A domain cannot be deleted as long as it has subordinate host objects, i.e. as long as a host object exists which name is a subdomain of the domain's name.
- A domain may have either 0 or 2-12 name servers. In the former case, it will be inactive and not published in the TLD zone.
- An authinfo string must have at least 4 characters and no more than 89 characters. No dictionary restrictions are imposed; registrars are responsible for using secure authinfo strings.
- Domains are auto-renewed by 1 year when they reach their expiration date + 45 days (autorenew grace period). The registrar is charged with the fee **after** the autorenew grace period, at which point the fee is non-refundable in case of domain deletion. **Note:** This auto-renewal will be performed even if the domain carries the status values "clientRenewProhibited" or "serverRenewProhibited", since these status values only prevent explicit renewal requests by the registrar. To prevent a domain from being auto-renewed, it must be deleted within 45 days after reaching its expiration date.
- A domain's validity period may not exceed 10 years at any given time.
- The add grace period is 5 days. For domains deleted within this period after creation, the registration fee is refunded. Also, such domains will not enter the redemption grace period but will immediately be released for re-registration. **Note:** This standard grace period may be different during certain registry launch phases if the policies in effect for these launch phases require a different add grace period duration. Please refer to the launch phase specifications for more information on launch phase specific add grace periods.
- The transfer grace period is 5 days. For domains deleted within this period after a transfer, the transfer fee is refunded.
- The renew grace period is 5 days. For domains deleted within this period after an explicit renewal, the renewal fee is refunded.
- The redemption grace period (RGP) is 30 days. Any domain deleted after the add grace period enters the redemption grace period, during which it may be restored by the previously sponsoring registrar.
- After restore, the RGP restore report must be submitted within 5 days. Otherwise, the domain will re-enter the redemption grace period (which is not restarted in this case).
- When the redemption grace period is over, the domain enters a "pre-release" period of a random length (but no longer than 5 days) during which it can only be restored via manual action by registry support staff. After that period, the domain is released for re-registration.
- Pending transfers are automatically acknowledged after 5 days (if the losing registrar does not reply earlier).
- The transfer block period (after domain creation or transfer) is 60 days. During this period, a domain may not be transferred to another registrar.
- DNSSEC data may be provisioned using the EPP extensions defined in RFC 5910, i.e. using the XML namespace "urn:ietf:params:xml:ns:secDNS-1.1". The obsolete XML namespace "urn:ietf:params:xml:ns:secDNS-1.0" is not supported.
- DNSSEC data must be provisioned using KeyData. DSData is not supported.
- Up to 4 DNSSEC KeyData entries are allowed per domain.

8.4.4.2 Domain Policies Specific to the .swiss TLD

8.4.4.2.1 Policy documents

The .swiss registration policy documents can be found on the registry web site http://nic.swiss (General Registration Policy, StartUp Plan Sunrise Dispute Resolution Policy, StartUp Plan Launch Period Policy).

The .swiss registration policy is based on the Federal Ordinance for Domain Names. The ordinance can be found online in three Swiss official languages. The English translation is published for convenience; in case of discrepancy the official German, French and Italian language versions are determinant.

- German: 784.104.2 Verordnung vom 5. November 2014 über Internet-Domains (VID)
  - https://www.admin.ch/opc/de/classified-compilation/20141744/index.html
- French: 784.104.2 Ordonnance du 5 novembre 2014 sur les domaines Internet (ODI)
  - https://www.admin.ch/opc/fr/classified-compilation/20141744/index.html
- Italian: 784.104.2 Ordinanza del 5 novembre 2014 sui domini Internet (ODIn)
  - https://www.admin.ch/opc/it/classified-compilation/20141744/index.html
- English: 784.104.2 Ordinance of 5 November 2014 on Internet Domains (OID) (translation)
  - https://www.admin.ch/opc/en/classified-compilation/20141744/index.html

8.4.4.2.2 Application paradigm is used at all times

The Federal Ordinance on Domain Names stipulates that a .swiss domain application must be published so as to enable other parties, who might have priority, to apply for the same .swiss domain name until the end of the publication period. This is a legal requirement applicable to .swiss both for the launch period and for general availability.

For this reason, the domain application paradigm is used not only during launch but also in general availability. All domain registration requests will receive the response code "1001 command completed successfully, action pending".

Multiple applications for the same domain name, or for variants thereof, constitute a contention set. Applications can be modified or withdrawn. If a contention set is not resolved by withdrawal of all but one application, the registry initiates a contention resolution process.

If only one validated application exists for a given domain (including variants) at the end of the publication period, that application will be allocated and become a registration.

8.4.4.2.3 Domain application validation

Domain application validation involves the validation of the registrant contact (if this has not already been done for another domain application) as well as a preliminary examination of whether the application complies with the rules, in particular eligibility and name selection rules. Multiple applications by different parties for the same domain may be validated.

Domain application validation in itself entails no presumption of allocation or entitlement. Domain applications may also be rejected because the domain name applied for is deemed to be a generic name of importance to the Swiss community, regardless whether that domain name string has previously been listed as such.

The rejection or withdrawal of an application triggers a full refund of the application fee from registry to registrar. However, the registry may deny a refund in case of abusive behavior.

The .swiss Registry may ask the registrar, the reseller or registrant to correct or complete the data. The registry may also set to correct validation-related attributes to facilitate validation.

8.4.4.2.4 News feed

The .swiss news feed may be used to carry announcements pursuant to the publication requirement for .swiss domain names. It may also carry other information of interest to the public for the .swiss TLD.

The news feed is based on ATOM (RFC 4287) on the URL http://whois.nic.swiss/feed . An announcement on the news feed may contain a link to the respective item on the Whois server.

8.4.4.2.5 Publication phase for Launch

All applications received during the launch phase are subject to a publication period ending 20 calendar days after the last domain applied for in the sunrise period has been validated and published. Until the end of those 20 days, any domain applied for during sunrise period may still be applied for by other parties.

8.4.4.2.6 Publication phases in General Availability

In general availability, applications are assigned to a publication phase at the moment their validation is completed. A new publication phase starts on a Wednesday at 13:00 UTC every week, ending 20 calendar days later at 13:00 UTC on a Tuesday.

8.4.4.2.7 Special procedures for name space mandates (NSMs)

Generic names may be allocated in the form of name space mandates (NSM). NSMs are described under Article 56 of the Federal Ordinance on Domain Names (see

- https://www.admin.ch/opc/de/classified-compilation/20141744/index.html#a56
- https://www.admin.ch/opc/fr/classified-compilation/20141744/index.html#a56
- https://www.admin.ch/opc/it/classified-compilation/20141744/index.html#a56).

A single NSM may comprise many domains. Applicants for NSMs must submit a proposal. Proposals for NSMs can be submitted in response to a Call for Proposals or spontaneously. Multiple domain names for a given purpose may be combined in a single mandate.

As the pricing for NSMs is on a per-mandate basis rather than a per-domain basis, the average price per domain invoiced may be considerably lower (or higher) than the price for standard domain registrations. The key criteria for selection for an NSM is best service to the community.

The mandate holder can select a registrar. The registrar can submit or take over NSM registrations after the negotiations between the registry and the mandate holder are complete, based on an exhaustive listing of domains belonging to the mandate and the use of authorization codes. A mandate may be subject to additions or removals of domain names.

8.4.4.2.8 Validation of Registrant Eligibility

The usage of a .swiss domain must comply with the registration agreement. To enable the registry to verify a registrant's eligibility for the use of a specific domain, registrars must supply an "enterprise ID" information string for the registrant contact of a .swiss domain, denoting the registered company ID of the company represented by the contact.

In case the registrar uses EPP, the string must be specified using the "http://xmlns.corenic.net/epp/contact-eligiblity-1.0" EPP extension upon creating or updating the domain's registrant contact; see the EPP extension's documentation further below in this document for more information. If the registrar uses the Control Panel, the string must be entered into a form field upon creating or updating the domain's registrant contact.

Note that contacts can be created, modified or deleted without using the extension. If an EPP contact update command does not explicitly change or remove the enterprise ID, the value will remain unchanged.

The enterprise ID is shown in the free-format section of the .swiss Whois output for domains, applications and contacts.

8.4.4.2.9 Finding the enterprise ID (UID/IDE/IDI) in official databases

The enterprise ID, in the specific context of .swiss based on current rules, is the Swiss UID/IDE/IDI. The full name is in German "Unternehmens-ID", in French "Numéro d’identification des entreprises (IDE)" and in Italian "Numero d’identificazione delle imprese (IDI)".

A Swiss UID/IDE/IDI always starts with "CHE" (the ISO-3166 alpha-3 code for Switzerland) followed by nine digits. A hyphen is generally used to separate "CHE" from the digits. The digits may be separated by dots into three groups of three digits. For the purpose of value-added tax, the Swiss UID/IDE/IDI often appears with the suffix "MWST", "TVA", "IVA" or "VAT". Organizations registered in the Swiss Commercial Register (Handelsregister / Registre du commerce / Registro di commercio) as well as government agencies have a unique Swiss UID/IDE/IDI.

It can be found online on the web site of the respective cantonal Register or on

- http://www.zefix.ch or
- https://www.uid.admin.ch .

The recommended format for the .swiss registry is "CHE-xxx.xxx.xxx" where each x is a digit.

8.4.4.2.10 Relationship between registrant eligibility and domain application validation

The string specified for the enterprise ID in the contact record will be validated by the registry as soon as it first appears in the registrant role for a domain application. The simple act of submitting an enterprise ID for a contact will not in itself trigger contact eligibility validation.

Contact eligibility validation is a component of domain application validation. Where appropriate, the .swiss registry contact and domain validation in a single workflow. When validation begins for the first domain application from a given prospective registrant, the registrant contact is validated to make sure the registrant is eligible and the enterprise ID is correct.

After a .swiss domain application has been validated, it is subject to the rules regarding publication and an opportunity for other parties to apply for the same domain name. Neither domain application validation nor contact eligibility validation thus constitute a right or a confirmation of domain registration.

Any subsequent domains registered by the same applicant will no longer require the component of contact eligibility validation, provided the same contact ID is attached to the domain application in the registrant role. A validated flag is set accordingly on the contact object and can be seen by the registrar using the EPP contact eligibility extension or via the control panel.

If the contact attached to a .swiss domain application in the registrant role does not have the enterprise ID set by way of the EPP contact eligibility extension, the registrar may set it through the .swiss registry Control Panel.

In order to facilitate validation, the .swiss registry may contact the registrar, or in consultation with the registrar, contact the reseller, the registrant and/or the administrative contact. The registry may set or correct the enterprise ID in consultation with the registrar.

It is possible that, for a given application, the registrant contact is validated as eligible whereas the domain application is rejected for reasons related to the applied-for domain name. In such a case, the contact record will remain in the validated state, unless there are reasons to subject it to re-validation.

• *8.4.5 IDN (Internationalized Domain Name) Policies*

The .swiss registry supports Internationalized Domain Names (IDNs). Provisioning IDNs via EPP requires the use of a special IDN EPP extension, for which documentation is provided in a separate document. The following IDN languages/scripts are supported:

- the Latin script

IDN variants are supported for some of the supported IDN languages. See the respective IDN table document for more information. For variants the so-called "attribute mode" is active, which means that all variants are attributes of some other domain. The number of variants per domain is limited to 10. The price for a domain with or without variants is the same, so variants are effectively for free.

• *8.5 Settings on EPP Test and Evaluation Systems*

The above settings are in place on a production instance of the CORE Registration System. However, to facilitate testing (especially in the field of grace periods and transfers), the following grace period settings differ from the production system on acceptance, staging and OT&E systems as follows:

- The add grace period is 5 minutes.
- The transfer grace period is 5 minutes.
- The renew grace period is 5 minutes.
- The redemption period is 5 minutes.
- The "pre-release" period is no longer than 5 minutes.
- After restore, the RGP restore report must be submitted within 5 minutes.
- The transfer block period (after domain creation or transfer) is 5 minutes.
- The time after which pending transfers are automatically acknowledged by the system is 5 minutes.

9. Launch Phase Specifications for the .swiss TLD

Prior to domain registrations within the .swiss TLD becoming generally available, the .swiss Registry conducts a number of launch phases. Each of these launch phases is defined by specific registrant eligibility criteria for domain name applications (or lack thereof). To provide proof of eligibility, registrants applying for domain names in a certain launch phase may have to provide (via their registrars) specific data that will be validated (synchronously or asynchronously) via the registry. This document describes the launch phases, their eligibility requirements, and the technical details about how data required for validation may be submitted to the registry via EPP.

9.1 Launch & GA Phases Overview

Upon registry startup, the .swiss TLD conducts the following launch phases:

1. "Sunrise": A standard sunrise phase during which certain registrants with trademarks registered in the TMCH (Trademark Clearinghouse) may apply for domain names that exactly match one of the labels from the TMCH SMD (Signed Mark Data
2. "Public Administrations": A launch phase during which Swiss Public Authorities and certain other entities may register names.
3. "Other Prior Rights": A launch phase allowing Swiss legal entities to apply for certain domain names, provided they are able to claim some prior right (trademark; corporate name, etc.).

Competing valid applications for the same domain name will be allocated based on the priority of the launch phase during which the application was submitted. The list above lists launch phases in order of descending priority, i.e. "Sunrise" has a higher priority than "Local Trademark" etc., which means that e.g. a "Sunrise" domain application wins over a "Other Prior Rights" application for the same name.

In general, EPP commands for domain applications during any of these launch phases are required to use the standard launch phase EPP extension specified at http://tools.ietf.org/html/draft-ietf-eppext-launchphase-02. Where applicable (see below), registrations may be subject to trademark claims, and proof of the respective claims notice must be specified via the mechanisms devised by this launch phase EPP extension.

ALL launch phases require an additionalInfo field describing the Intended Use for the domain name, this includes the two "General Availability" phases after the end of the Launch Phase:

1. "Claims": A general availability phase during which TMCH trademark claims will be checked by the registry.
2. "Open": The general availability phase following the "Claims" phase. No TMCH trademark claims are checked in this final phase, which continues indefinitely. Some Launch Phases require specific info to be provided in such field for validation purposes.

Even in GA phases, all applications must be approved by the Registry according to the Eligibility and Name Selection criteria set in the .swiss General Registration Policy. And all validated names will be published for a period of 20 days, allowing contending applications to be filed during such period, prior to turning into registrations. Please refer to .swiss Policies for specific details.

9.1.1 "Sunrise" Launch Phase

The "Sunrise" launch phase (end-date Sunrise) requires registrants to supply a TMCH-signed set of mark data, i.e. a valid SMD file. One of the labels within the SMD must be a literal match for the second level domain name to be registered. Registrations in this phase are not subject to trademark claims notices (since providing the SMD already proves the trademark holder's awareness of the registration).

Please note than the SMD must either:

- correspond to a Swiss trademark (jurisdiction=ch or wipo, with legal effects in Switzerland), or
- corresponding to a non-Swiss trademark, the holder of such trademark must be a Swiss entity.

In EPP, the required data must be specified via a standard <launch:create> EPP extension element carrying signed mark data according to http://tools.ietf.org/html/draft-ietf-eppext-launchphase-02, using the standard launch phase identifier "sunrise".

IN addition, the registration must be submitted specifying the following mandatory <applicationInfo> field:

- intended-use

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.swiss</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.irondns.net</domain:hostObj>
          <domain:hostObj>ns2.irondns.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>CONTACT-1</domain:registrant>
        <domain:contact type="admin">CONTACT-1</domain:contact>
        <domain:contact type="tech">CONTACT-1</domain:contact>
        <domain:contact type="billing">CONTACT-1</domain:contact>
        <domain:authInfo>
          <domain:pw>zS2fB#CR[1QgI#$0</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>sunrise</launch:phase>
        <ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
          <smd:encodedSignedMark xmlns:smd="urn:ietf:params:xml:ns:signedMark-1.0">
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHNtZDpzaWduZWRNYXJrIHht
bG5zOnNtZD0idXJuOmlldGY6cGFyYW1zOnhtbDpuczpzaWduZWRNYXJrLTEuMCIgaWQ9Il85MGNm
...
dGlmaWNhdGU+PC9kczpYNTA5RGF0YT48L2RzOktleUluZm8+PC9kczpTaWduYXR1cmU+PC9zbWQ6
c2lnbmVkTWFyaz4=
          </smd:encodedSignedMark>
          <!-- mandatory elements -->
          <ext:applicationInfo
            type="intended-use">Statement-of-intended-use-for-domain-name</ext:applicationInfo>
        </ext:augmentedMark>
      </launch:create>
    </extension>
  </command>
</epp>
```

9.1.2 "Public Administrations" Launch Phase

The "Public Administrations" launch phase (end-date Limited Registration Period) allows Swiss Public Authorities and certain other organizations uner Public law to register certain domain names.

The registration must be submitted specifying the following mandatory <applicationInfo> field:

- intended-use

Registrations in this phase may be subject to trademark claims notices.

In EPP, the required data must be specified via a standard <launch:create> EPP extension element according to http://tools.ietf.org/html/draft-ietf-eppext-launchphase-02, using the custom launch phase identifier "public-interest". As mark data, an element of type <augmentedMark> (as defined in the document "XML Mark Data Extension for the .swiss TLD") must be specified, which must contain <applicationInfo> element with type "intended-use".

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.swiss</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.irondns.net</domain:hostObj>
          <domain:hostObj>ns2.irondns.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>CONTACT-1</domain:registrant>
        <domain:contact type="admin">CONTACT-1</domain:contact>
        <domain:contact type="tech">CONTACT-1</domain:contact>
        <domain:contact type="billing">CONTACT-1</domain:contact>
        <domain:authInfo>
          <domain:pw>zS2fB#CR[1QgI#$0</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="public-administrations">custom</launch:phase>
        <ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
          <!-- mandatory elements -->
          <ext:applicationInfo
            type="intended-use">Statement-of-intended-use-for-domainanme</ext:applicationInfo>
        </ext:augmentedMark>
      </launch:create>
    </extension>
  </command>
</epp>
```

9.1.3 "Other Prior Rights" Launch Phase

The "Other Prior Rights" launch phase (end-date Limited Registration Period) requires registrants to supply evidence/reference that they are holders of a legitimate entitlement to the name, such as valid, non-TMCH Swiss trademark, a Swiss designation of source of geographical name (as defined by Swiss law) or the corporate name of the applicant, which makes them eligible for registering the domain name. No TMCH trademark data (SMD) needs to be supplied,

The registration must be submitted specifying the following mandatory <applicationInfo> fields:

- intended-use: The Statement of Intended Use for the domain name, including explicit reference to the applicants' claimed right to the name (if not the corporate name of the applicant). For example, if it matches a Swiss tramark, the TM number must be provided.

Registrations in this phase may be subject to trademark claims notices.

In EPP, the required data must be specified via a standard <launch:create> EPP extension element according to http://tools.ietf.org/html/draft-ietf-eppext-launchphase-02, using the custom launch phase identifier "local-trademark". As mark data, an element of type <augmentedMark> (as defined in the document "XML Mark Data Extension for the .swiss TLD") must be specified, which must contain <applicationInfo> elements with types "trademark-id", "trademark-issuer" and "intended-use".

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.swiss</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.irondns.net</domain:hostObj>
          <domain:hostObj>ns2.irondns.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>CONTACT-1</domain:registrant>
        <domain:contact type="admin">CONTACT-1</domain:contact>
        <domain:contact type="tech">CONTACT-1</domain:contact>
        <domain:contact type="billing">CONTACT-1</domain:contact>
        <domain:authInfo>
          <domain:pw>zS2fB#CR[1QgI#$0</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="other-rights">custom</launch:phase>
        <ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
          <!-- mandatory elements -->
          <ext:applicationInfo
            type="intended-use">Statement-of-intended-use-for-domainanme</ext:applicationInfo>
        </ext:augmentedMark>
      </launch:create>
    </extension>
  </command>
</epp>
```

9.1.4 "Claims" Phase

The "Claims" phase, during the first 90 days of General Availability, does require specifying the following mandatory <applicationInfo> field:

- intended-use

Registrations in this phase may be subject to trademark claims notices.

Also, in EPP, the standard <launch:create> EPP extension according to http://tools.ietf.org/html/draft-ietf-eppext-launchphase-02 needs to be used in order to specify the standard launch phase identifier "claims". As mark data, an element of type <augmentedMark> (as defined in the document "XML Mark Data Extension for the .swiss TLD") must be specified, which must contain <applicationInfo> elements with type "intended-use".

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.swiss</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.irondns.net</domain:hostObj>
          <domain:hostObj>ns2.irondns.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>CONTACT-1</domain:registrant>
        <domain:contact type="admin">CONTACT-1</domain:contact>
        <domain:contact type="tech">CONTACT-1</domain:contact>
        <domain:contact type="billing">CONTACT-1</domain:contact>
        <domain:authInfo>
          <domain:pw>zS2fB#CR[1QgI#$0</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>claims</launch:phase>
        <ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
          <!-- mandatory elements -->
           <ext:applicationInfo
            type="intended-use">Statement-of-intended-use-for-domain-name</ext:applicationInfo>
        </ext:augmentedMark>
      </launch:create>
    </extension>
  </command>
</epp>
```

9.1.5 "Open" Phase

The "Open" phase is the General Availability once the mandatory Claims Notices period has expired, does require specifying the following mandatory <applicationInfo> field:

- intended-use

Registrations in this phase are not subject to trademark claims notices.

Also, in EPP, the standard <launch:create> EPP extension according to http://tools.ietf.org/html/draft-ietf-eppext-launchphase-02 needs to be used in order to specify the standard launch phase identifier "open". As mark data, an element of type <augmentedMark> (as defined in the document "XML Mark Data Extension for the .swiss TLD") must be specified, which must contain <applicationInfo> elements with type "intended-use".

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.swiss</domain:name>
        <domain:period unit="y">1</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.irondns.net</domain:hostObj>
          <domain:hostObj>ns2.irondns.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>CONTACT-1</domain:registrant>
        <domain:contact type="admin">CONTACT-1</domain:contact>
        <domain:contact type="tech">CONTACT-1</domain:contact>
        <domain:contact type="billing">CONTACT-1</domain:contact>
        <domain:authInfo>
          <domain:pw>zS2fB#CR[1QgI#$0</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase>open</launch:phase>
        <ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
          <!-- mandatory elements -->
           <ext:applicationInfo
            type="intended-use">Statement-of-intended-use-for-domain-name</ext:applicationInfo>
        </ext:augmentedMark>
      </launch:create>
    </extension>
  </command>
</epp>
```

10. XML Mark Data Extension for specifying augmented trademark information during Launch Phases

The CORE Registration System used to operate the .swiss Registry fully supports the standard launch phase EPP extension (as defined at http://tools.ietf.org/html/rfc8334.txt) for the submission of launch phase and trademark information during a registry's launch phases. However, while this standard extension does offer sufficient support for ordinary landrush, TMCH sunrise and TMCH claims based launch phases, it does not provide means to collect any additional information from registrars/registrants if the need arises. For this purpose, the CORE Registration System provides support for "augmented" mark data via custom XML element usable as mark data within the standard launch phase extension. Details about this augmented mark data are provided in this document.

10.1 Introduction

Some registries may require registrants (and, in turn, their registrars) to provide additional information during certain launch phases and submit this information along with standard domain data in an EPP request. The launch phase EPP extension (specified at http://tools.ietf.org/html/rfc8334.txt) provides standardized support for the most common requirements, i.e. for specifying the launch phase itself, plus (signed or unsigned) trademark data. In some situations, this may not be sufficient. For example, a registry for a TLD representing a certain geographical region may choose to devise a launch phase during which local public authorities shall get a chance to register certain names; such a launch phase may require the specification of additional information needed to validate a domain application, such as e.g. a description of the public authority, a URL where the public authority may be validated, or a reference URL at which an existing public online presence of the authority may be found.

To accommodate such requirements, the CORE Registration System supports an extension of the mark data definitions used by the standard launch phase extension that allows the specification of arbitrary key-value data (plus free text information) alongside standard mark data. This document describes how this mark data extension is represented in EPP/XML.

10.2 Implementation Details

In the following, the XML name space prefixes "mark" and "smd" shall refer to the "urn:ietf:params:xml:ns:mark-1.0" and "urn:ietf:params:xml:ns:signedMark-1.0" name spaces, both as specified at http://tools.ietf.org/html/rfc7848.

This extension for augmented mark data utilizes the extension mechanism devised by the standard launch phase extension, which is based on XSD substitution groups. Specifically, the <augmentedmark> element defined by this extension belongs to the substitution group "smd:abstractSignedMark" and uses an XML schema type that extends the "smd:abstractSignedMarkType" type defined by the launch phase extension. Consequently, the <augmentedMark> element may substitute, i.e. occur in place of (non-augmented) standard mark data within the launch phase extension. Practically, this means that a registrar must simply use an <augmentedMark> element instead of standard mark within the launch phase EPP extension whenever a launch phase requires mandatory (or allows optional) augmented data; the actual mark data may be used (unaltered) as an inner element within the <augmentedMark> element (see below).

Note: this means that the augmented mark data is not implemented via a "top-level" EPP extension, but is merely utilizing an extension mechanism that allows extending the types of mark data accepted by the existing launch phase extension. This approach should facilitate the adoption by registrars using EPP.

The content of the <augmentedMark> element is structured as follows:

- First, an (optional) element containing the actual mark data may be present; this may either be a <mark:abstractMark>, <smd:abstractSignedMark> or <smd:encodedSignedMark> element. This element is optional, allowing the augmented mark to be used as a pure key-value based data set if a launch phase does not require actual mark data.
- Second, one or more <applicatonInfo> elements must follow which contain the additional information (the value). An optional "type" element may indicate the key for which the value supplied data. If the "type" element is omitted, the content of the <applicationInfo> element represents free-text information with no specific semantics. Note that the same value for the "type" element may not occur more than once; also, no more than one <applicationInfo> with the "type" element omitted may occur. The order in which the <applicationInfo> elements occur is irrelevant.

Note: the <augmentedMark> element provides a generic way to express free-text and key/value based data. The actual types (keys) and values required/allowed in <applicationInfo> elements within an <augmentedMark> element, as well as the requirement for mark data, are defined by registry policy and may vary between different launch phases of the same registry. Please refer to the respective registry's policy documentation for more information on the required/optional <applicationInfo> type attributes and respective expected element content.

10.3 Examples

The following <augmentedMark> example specifies standard encoded mark data (as received from the TMCH) and augments it with a free-text <applicationInfo> plus two <applicationInfo> elements (with types "example-type-1" and "example-type-2":

```
<ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
  <smd:encodedSignedMark xmlns:smd="urn:ietf:params:xml:ns:signedMark-1.0">
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHNtZDpzaWduZWRNYXJrIHht
bG5zOnNtZD0idXJuOmlldGY6cGFyYW1zOnhtbDpuczpzaWduZWRNYXJrLTEuMCIgaWQ9Il82YzMw
...
VnFhUT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEYXRhPjwvZHM6S2V5SW5mbz48L2Rz
OlNpZ25hdHVyZT48L3NtZDpzaWduZWRNYXJrPg==
  </smd:encodedSignedMark>
  <ext:applicationInfo>Arbitrary text goes here.</ext:applicationInfo>
  <ext:applicationInfo type="example-type-1">Example Content</ext:applicationInfo>
  <ext:applicationInfo type="example-type-2">other example content</ext:applicationInfo>
</ext:augmentedMark>
```

In contrast, the following <augmentedMark> example contains a single, free-text <applicationInfo> element and omits mark data entirely:

```
<ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
  <ext:applicationInfo>Arbitrary text goes here.</ext:applicationInfo>
</ext:augmentedMark>
```

Finally, the following XML document represents a full <domain:create> EPP command that utilizes the <augmentedMark> from the first example above to submit augmented mark data for an application during the "somelaunchphase" launch phase:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <command>
    <create>
      <domain:create xmlns:domain="urn:ietf:params:xml:ns:domain-1.0">
        <domain:name>example.swiss</domain:name>
        <domain:period unit="y">2</domain:period>
        <domain:ns>
          <domain:hostObj>ns1.irondns.net</domain:hostObj>
          <domain:hostObj>ns2.irondns.net</domain:hostObj>
        </domain:ns>
        <domain:registrant>CONTACT-1</domain:registrant>
        <domain:contact type="admin">CONTACT-1</domain:contact>
        <domain:contact type="tech">CONTACT-1</domain:contact>
        <domain:contact type="billing">CONTACT-1</domain:contact>
        <domain:authInfo>
          <domain:pw>zS2fB#CR[1QgI#$0</domain:pw>
        </domain:authInfo>
      </domain:create>
    </create>
    <extension>
      <launch:create xmlns:launch="urn:ietf:params:xml:ns:launch-1.0">
        <launch:phase name="somelaunchphase">custom</launch:phase>
        <ext:augmentedMark xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0">
          <smd:encodedSignedMark xmlns:smd="urn:ietf:params:xml:ns:signedMark-1.0">
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPHNtZDpzaWduZWRNYXJrIHht
bG5zOnNtZD0idXJuOmlldGY6cGFyYW1zOnhtbDpuczpzaWduZWRNYXJrLTEuMCIgaWQ9Il82YzMw
...
VnFhUT08L2RzOlg1MDlDZXJ0aWZpY2F0ZT48L2RzOlg1MDlEYXRhPjwvZHM6S2V5SW5mbz48L2Rz
OlNpZ25hdHVyZT48L3NtZDpzaWduZWRNYXJrPg==
          </smd:encodedSignedMark>
          <ext:applicationInfo>Arbitrary text goes here.</ext:applicationInfo>
          <ext:applicationInfo type="example-type-1">Example Content</ext:applicationInfo>
          <ext:applicationInfo type="example-type-2">other example
content</ext:applicationInfo>
        </ext:augmentedMark>
      </launch:create>
    </extension>
  </command>
</epp>
```

10.4 Formal Syntax

This XML schema definition provides a formal description of the XML structures defined by this extension:

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://xmlns.corenic.net/epp/mark-ext-1.0"
  xmlns:ext="http://xmlns.corenic.net/epp/mark-ext-1.0"
  xmlns:mark="urn:ietf:params:xml:ns:mark-1.0"
  xmlns:smd="urn:ietf:params:xml:ns:signedMark-1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <annotation>
    <documentation>
      extension schema for the submission of additional
      launch phase/application data if standard mark data
      is not sufficient; the schema defines an "augmentedMark" element
      which type allows it to be used in place of standard mark data
      in the launch phase extension (http://tools.ietf.org/html/rfc7848)
      and to augment mark data with arbitrary, key-value style additional
      information required for evaluating domain name applications
    </documentation>
  </annotation>


  <!-- import mark data definitions -->

  <import namespace="urn:ietf:params:xml:ns:mark-1.0"
    schemaLocation="mark-1.0.xsd"/>

  <import namespace="urn:ietf:params:xml:ns:signedMark-1.0"
    schemaLocation="signedmark-1.0.xsd"/>


  <!-- type for application info values -->

  <simpleType name="applicationInfoValueType">
    <restriction base="normalizedString">
      <minLength value="1"/>
      <maxLength value="2048"/>
    </restriction>
  </simpleType>


  <!-- type for application info types -->

  <simpleType name="applicationInfoTypeType">
    <restriction base="token">
      <minLength value="1"/>
      <maxLength value="64"/>
    </restriction>
  </simpleType>


  <!-- type for application info data -->

  <complexType name="applicationInfoType">
    <simpleContent>
      <extension base="ext:applicationInfoValueType">
        <!-- optional attribute to denote the type of information provided -->
        <attribute name="type" type="ext:applicationInfoTypeType"
          use="optional"/>
      </extension>
    </simpleContent>
  </complexType>


  <!-- sub-type of abstractSignedMarkType suitable for
       use in the launch phase extension -->

  <complexType name="augmentedMarkType">
    <complexContent>
      <extension base="smd:abstractSignedMarkType">
        <sequence>

          <!-- optional embedded standard mark data;
               unsigned, signed or encoded -->
          <choice minOccurs="0">
            <element ref="mark:abstractMark"/>
            <element ref="smd:abstractSignedMark"/>
            <element ref="smd:encodedSignedMark"/>
          </choice>

          <!-- additional information for a domain application;
               at least one must be provided -->
          <element name="applicationInfo" type="ext:applicationInfoType"
            maxOccurs="unbounded"/>

        </sequence>
      </extension>
    </complexContent>
  </complexType>


  <!-- top-level element for augmented mark data -->

  <element name="augmentedMark" type="ext:augmentedMarkType"
    substitutionGroup="smd:abstractSignedMark"/>

</schema>
```

11. CDS/CDNSKEY

The CORE Registration System supports managing DS records from the parent via CDS/CDNSKEY records placed in child zones, as specified in RFC 8078 (https://www.rfc-editor.org/rfc/rfc8078). Since DNSSEC data must be specified using DNSKEY records in the CORE Registration System (rather than by DS records), only CDNSKEY records are supported; CDS records will not be considered. Also, this feature requires the domain's zone to be hosted by a DNS operator that supports CDS/CDNSKEY.

CDNSKEY allows the operator of the a domain's zone to inform the parent about changes. In particular, it is possible to

- Set up DNSSEC
- Rollover the DNSKEY

- Deactivate DNSSEC

11.1 Acceptance

For an initially unsigned domain, the CORE Registration System offers two possibilities to activate DNSSEC.

11.1.1 Accept from Inception

Directly after a domain is created, the domain's zone is looked up at the configured name servers. If CDNSKEY records are found in the zone, these are taken to set the domain's DNSKEY records in the CORE Registration System. This possibility only exists for the first few minutes after the domain creation. It should therefore be ensured that the corresponding zone is correctly set up before the domain is registered.

11.1.2 Accept with Authenticated Signals from the Zone's Operator

This feature is based on the draft-ietf-dnsop-dnssec-bootstrapping-02 (https://www.ietf.org/id/draft-ietf-dnsop-dnssec-bootstrapping-02.html). In order to use this acceptance feature, the domain's zone must be hosted by a DNS operator supporting the bootstrapping mechanism. Contrary to the previously described "Accept from Inception", this feature can also be used for already registered domains.

11.2 DNSKEY Rollover

Once the domain is successfully signed, any DNSKEY rollovers initiated by the DNS operator (and reflected within the CDNSKEY record set) will automatically be detected and the domain will be updated accordingly in the CORE Registration System.

11.3 Deactivate DNSSEC

By providing an appropriate CDNSKEY record (see Section 4 of RFC 8078 (https://www.rfc-editor.org/rfc/rfc8078#section-4)) the DNS operator can signal to deactivate DNSSEC.

11.4 Poll Messages

For all the changes mentioned above, the CORE Registration System injects "domain update" poll messages according to RFC 8590, "Change Poll Extension" (https://tools.ietf.org/html/rfc8590) into the affected domain's registrar's poll message queue.

Example Poll Response for a domain's DNSSEC initialization:

```
<?xml version="1.0" encoding="UTF-8"?>
<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
  <response>
    <result code="1301">
      <msg>Command completed successfully; ack to dequeue</msg>
    </result>
    <msgQ count="3" id="81">
      <qDate>2023-01-12T12:14:16.040Z</qDate>
    </msgQ>
    <resData>
      <infData xmlns="urn:ietf:params:xml:ns:domain-1.0">
        <name>cds-test.tld</name>
        <roid>nhgw341-D</roid>
        <status s="ok"/>
        <clID>reg-411847</clID>
        <crID>reg-411847</crID>
        <crDate>2023-01-12T10:07:34.595Z</crDate>
        <upID>REGISTRY</upID>
        <upDate>2023-01-12T12:14:16.040Z</upDate>
        <exDate>2024-01-12T10:07:34.595Z</exDate>
      </infData>
    </resData>
    <extension>
      <infData xmlns="urn:ietf:params:xml:ns:secDNS-1.1">
        <keyData>
          <flags>257</flags>
          <protocol>3</protocol>
          <alg>13</alg>
          <pubKey>
```

```
HfgIZ9KH+0rALsAQNkbG9CJ+VkhNZKCyFOVYTqpBxemt5wuEJNC5q/1IjBHlUaPAgzfpJLOn0rv7TyYB444xsg==
          </pubKey>
        </keyData>
      </infData>
      <changeData xmlns="urn:ietf:params:xml:ns:changePoll-1.0" state="after">
        <operation>update</operation>
        <date>2023-01-12T12:14:16.040Z</date>
        <svTRID>CP-230112110131-4</svTRID>
        <who>CDS/CDNSKEY</who>
        <reason>DNSSEC initialized</reason>
      </changeData>
    </extension>
    <trID>
      <svTRID>1673856943051-2</svTRID>
    </trID>
  </response>
</epp>
```

12. Measures for GDPR Compliance

Generally, for auditing purposes, the CORE Registration System retains all historical data of registered objects (i.e., all current and previous versions of contacts, hosts, domains, applications), as well as information about all persistent orders, i.e. orders which modified the data of registered objects.

However, in order to comply with the European Union's General Data Protection Regulation (GDPR), the CORE Registration System features a mechanism which automatically removes obsolete data potentially containing personal information from its database. This process exclusively affects data which is no longer vital for proper operation of the registry.

The concrete measures implemented by the mechanism are described in the following sections.

12.1 Deletion of Old, Unused Contacts

All active (non-deleted) contacts which have not been modified for more than **one year** and have also not been referenced in any domains or applications for more than **one year** are automatically deleted from the system as part of a general cleanup mechanism. In order to inform the registrars about this unsolicited deletion of their unused contact objects, the system injects poll messages according to RFC 8590, "Change Poll Extension" (https://tools.ietf.org/html/rfc8590) into the registrar's poll message queue for every deleted contact.

12.2 Removal of Deleted Contacts

All deleted contacts which were deleted more than **two years** ago are automatically removed from the system, along with all their versions which existed before their deletion. This includes contacts deleted due to the contact cleanup mechanism described above.

12.3 Removal of Old Contact Versions

All old contact versions whose validity range ended more than **two years** ago are automatically removed from the system (even if the contact itself is not deleted, i.e. still active). Affected active contacts remain fully usable, but their version history is truncated.

12.4 Removal of Old, Deleted Domains

All finally deleted (i.e., non-restorable) domains and applications which were deleted more than **two years** ago are automatically removed from the system, along with all their versions which existed before their deletion.

12.5 Removal of Old Orders

All persistent orders containing personal data (e.g., contact creations or contact modifications) which have been processed more than **two years** ago are automatically removed from the system. In addition, any orders associated with objects or object versions removed due to the other GDPR measures listed above are also removed.

• 13. Provisions for Unhandled Namespaces in Poll Messages

Whenever the CORE Registration System delivers a poll message to the registrar's EPP client, it checks whether the poll message would contain EPP extension XML elements with XML namespaces for which the registrar has not announced support in the EPP <login> command that created the current EPP session. If such XML elements are detected, the CORE Registration System follows the operational practice recommended by the IETF's Unhandled Namespaces RFC (http://tools.ietf.org/html/rfc9038.txt), in order to not inject a "poisonous" poll message that may break or block the registrar's poll queue processing, e.g. if the registrar's EPP client performs XML schema validation on server responses.

For example, if a poll message is supposed to contain a "contact purged" message with a "change poll" extension, the normal poll message would look like this:

```
<?xml version="1.0" encoding="UTF-8"?><epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
      </result>
      <msgQ count="13" id="1028">
         <qDate>2020-05-19T14:24:37.444Z</qDate>
      </msgQ>
      <resData>
         <infData xmlns="urn:ietf:params:xml:ns:contact-1.0">
            <id>C4405-T</id>
            <roid>cpdt2925-C</roid>
            <status s="ok"/>
            <postalInfo type="int">
               <name>Joe Average</name>
               <org>Acme</org>
               <addr>
                  <street>Elm Street 9</street>
                  <city>Springfield</city>
                  <sp>NRW</sp>
                  <pc>12345</pc>
                  <cc>DE</cc>
               </addr>
            </postalInfo>
            <postalInfo type="loc">
               <name>Joe Average</name>
               <org>Acme</org>
               <addr>
                  <street>Elm Street 9</street>
                  <city>Springfield</city>
                  <sp>NRW</sp>
                  <pc>12345</pc>
                  <cc>DE</cc>
               </addr>
            </postalInfo>
            <voice>+49.232232211</voice>
            <fax>+49.232232311</fax>
            <email>joe.average@acme.com</email>
            <clID>reg-411847</clID>
            <crID>reg-411847</crID>
            <crDate>2019-10-02T12:12:17.166Z</crDate>
            <authInfo>
               <pw>some-password</pw>
            </authInfo>
         </infData>
      </resData>
      <extension>
         <changeData xmlns="urn:ietf:params:xml:ns:changePoll-1.0" state="before">
            <operation>autoPurge</operation>
            <date>2020-05-19T14:24:37.444Z</date>
            <svTRID>1589898277444-1</svTRID>
            <who>registry</who>
            <reason>contact old and unused</reason>
         </changeData>
      </extension>
      <trID>
         <clTRID>poll-1614791605927</clTRID>
         <svTRID>1614791605929-1428</svTRID>
      </trID>
   </response>
</epp>
```

This poll message would only be delivered in this fashion (with the <changeData> element present in the response's <extension> element) if the registrar's <login> command contained the change poll extensions XML namespace URI ("urn:ietf:params:xml:ns:changePoll-1.0") in an <extURI> element within its <svcExtension> section. By including it, the registrar announces that its client system supports the change poll extension, and the CORE Registration System assumes that the client is able to handle the extension's XML elements.

However, if no such support was announced by the registrar in the <login> command, the CORE Registration System includes the <changeData> element in a <value> element within an <extValue> element in the poll response's result instead, like this:

```
<?xml version="1.0" encoding="UTF-8"?><epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
   <response>
      <result code="1301">
         <msg>Command completed successfully; ack to dequeue</msg>
         <extValue>
            <value>
               <changeData xmlns="urn:ietf:params:xml:ns:changePoll-1.0" state="before">
                  <operation>autoPurge</operation>
                  <date>2020-05-19T14:24:37.444Z</date>
                  <svTRID>1589898277444-1</svTRID>
                  <who>registry</who>
                  <reason>contact old and unused</reason>
               </changeData>
            </value>
            <reason>urn:ietf:params:xml:ns:changePoll-1.0 not in login services</reason>
         </extValue>
      </result>
      <msgQ count="13" id="1028">
         <qDate>2020-05-19T14:24:37.444Z</qDate>
      </msgQ>
      <resData>
         <infData xmlns="urn:ietf:params:xml:ns:contact-1.0">
            <id>C4405-T</id>
            <roid>cpdt2925-C</roid>
            <status s="ok"/>
            <postalInfo type="int">
               <name>Joe Average</name>
               <org>Acme</org>
               <addr>
                  <street>Elm Street 9</street>
                  <city>Springfield</city>
                  <sp>NRW</sp>
                  <pc>12345</pc>
                  <cc>DE</cc>
               </addr>
            </postalInfo>
            <postalInfo type="loc">
               <name>Joe Average</name>
               <org>Acme</org>
               <addr>
                  <street>Elm Street 9</street>
                  <city>Springfield</city>
                  <sp>NRW</sp>
                  <pc>12345</pc>
                  <cc>DE</cc>
               </addr>
            </postalInfo>
            <voice>+49.232232211</voice>
            <fax>+49.232232311</fax>
            <email>joe.average@acme.com</email>
            <clID>reg-411847</clID>
            <crID>reg-411847</crID>
            <crDate>2019-10-02T12:12:17.166Z</crDate>
            <authInfo>
               <pw>some-password</pw>
            </authInfo>
         </infData>
      </resData>
      <trID>
         <clTRID>poll-1614791550928</clTRID>
         <svTRID>1614791550931-1389</svTRID>
      </trID>
   </response>
</epp>
```

As the EPP RFC defines the content of that <value> element as unparsed XML content of any namespace, even validating EPP clients should not run into any problems processing such messages.

However, it is recommended that registrars monitor their EPP poll messages for occurrences of the "not in login services" hint in any result <reason> elements, as such poll messages may contain important operational information that may otherwise go unnoticed.

14. Extension for Language Information

The CORE Registration System used to operate the .swiss TLD provides a proprietary EPP extension for internationalized domain names (IDNs).

14.1 Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) described in RFC 5730. This mapping is an extension of the domain name mapping described in RFC 5731. It is specified using the Extensible Markup Language (XML) and XML Schema notation.

This extension serves the purpose of supplying and querying information for internationalized domain names. In particular, the language or script used and domain name variants are addressed.

The CORE Registration System is capable with dealing the needs of the various registries by using different modes of operation. Each mode, however, requires slightly different input from the registrars and generate slightly different output to the registrars. In order to avoid a multitude of similar extensions, a unified extension has been created to cover all cases. In the following, all supported concepts are explained, although not all are relevant for the .swiss TLD. Specific notes indicate whether they apply to the .swiss TLD or not. The examples have been chosen in such a way that only relevant parts are contained.

The following subsections summarize details required for the understanding of the IDN extension. For further policy details, see "IDN Policies" and "Variant Policies" sections above.

14.1.1 Domain Names

Domain names consists of a sequence of characters of the Unicode character set. If the domain name uses only characters of the ASCII subset, it is called an ASCII domain name, otherwise, it is called an internationalized domain name, abbreviated as IDN. Registrable ASCII domain names are further limited to letters, digits and hyphens (abbreviated LDH), so they cannot contain symbols (like the percent sign) or punctuation (like the exclamation mark).

IDNs must generally conform to the IDNA2008 standard. For registrable names, further language and script restrictions apply. Within EPP, IDNs have to be encoded by the Punycode standard (typically identifiable by the "xn--" prefix), including, but not limited to the IDN extension.

14.1.2 Language and Script

ICANN requires that the registration of an IDN must be accompanied by the specification of a language or script. A "script" denotes a writing system, like the use of Latin, Cyrillic or Arabic characters or Asian ideographs. A "language" denotes in this context the use of a writing system for a specific human language. With the selection of a script or language, the characters that can be used to form a domain name are limited to those used by the respective script or language. The exact characters and rules are documented in the so-called IDN tables, published by the registries and the IANA (http://www.iana.org/domains/idn-tables), the tables applicable for the .swiss TLD are included further down in this documentation. The reason behind this is to mitigate the risks of so-called homonym attacks, which use identical or similar looking characters from different scripts or languages to spoof the users about the real identity of a domain name.

To specify the language or script, the extension provides the <lang> or <script> elements, respectively. The language identifier meets the requirements of RFC 5646 (which is based on the ISO 639-1 language identifiers), while the script identifier meets the requirements of ISO 15924. While client provided language and script identifiers are accepted in a case insensitive manner, they are always reported in the correct case. The .swiss TLD uses the <script> element.

The following IDN tags are supported by the .swiss TLD:

Name	tag	type	supports variants	default
none/ASCII	*(empty)*	none		x
Latin	Latn	script		

14.1.3 Variants

In some languages, there is more than one way to write a word. In Latin languages, accented characters can often be replaced by their base characters or ligatures can be replaced by the letters from which they derive, which was common practice back when mechanical typewriters and early computers could not represent such characters. It is still important today as not all protocols and systems are ready to deal with IDNs. For other languages in other scripts, similar replacements exists. For example, the Chinese differentiate between "traditional" and "simplified" ideographs, so that for many words multiple ideographs exist. The different ways to write a word needs to be distinguished from synonyms – two completely different words (often of different origins) meaning the same or something similar. Usually this is not covered by variants.

The variants mechanism is also a way to mitigate the homograph problem. The registry system does not allow two names, which are variants of each other, to be registered by two different registrants. In order to achieve this, the registry system supports two modes: The first is called "attribute mode", the second is called "object mode".

In the "attribute mode", for a registered domain name, variant names can be added and removed. These variant names are an attribute of the domain object, i.e. integral data of it, comparable to the contact and hosts references, the DNSSEC data and the status values. All variants are published in the DNS using the same name server set and DNSSEC keys as the original domain name.

In the "object mode", each variant is a separate domain object. To create a new variant, a new domain:create EPP command must be issued. To prevent the registration by a different registrant, such a variant may only be created by the same registrar and the variant must share some attributes like registrant, administrative contact and/or name servers with the existing domain name. However, the variant may have different other contacts, may use other name servers and other DNSSEC data. The variants and the original domain name form a so-called *bundle*. The original domain name has, contrary to the "attribute mode", no special role. For example, if the original domain is deleted, the variants survive. Changes to the registrant (and administrative) contacts applied to one domain of the bundle are mirrored to all other domains of the bundle. Similarly, if a transfer on one domain is performed, the other domains of the bundle are transferred as well.

Throughout the extensions, the domain names of variants are represented by <nameVariant> elements, each containing one name in Punycode notation, as noted above. The <nameVariant> elements are wrapped in a single <variants> element for all commands and response extensions except the <idn:update> extension. An omitted <variants> element has the same meaning as a <variants> element containing no variants. For the <idn:update> extension, the <nameVariant> elements directly appear within the <add> and <rem> elements. There is no special order defined on variants, they may be submitted or reported in an arbitrary order.

The .swiss TLD supports variants in the "attribute mode".

Per domain object, up to 10 variants may be added.

14.2 EPP Command Mapping

This section deals with the specific command mappings for the .swiss TLD EPP extension for IDNs.

In the following, the respective root elements of the extensions are mentioned. If used, they must be placed or expected within the optional <extension> element at the proper location in the XML document representing the EPP command or response, as described in RFC 5730. Note that the use of the "idn:" XML namespace prefix is for documentation purposes only. Conforming to the "Namespaces in XML 1.1" standard, EPP and the registry implementation take only the associated namespace URI into account, and not the prefix itself. So actually any prefix or even the default namespace may be used in requests and must be expected in responses.

The IDN extension is only used in relation to domain objects. It will not occur in commands that are related to host and contact objects.

14.2.1 EPP Query Commands

There are four EPP commands to retrieve object information: <check> to find out whether an object is known to the server, <info> to ask for detailed information associated with an object, <poll> to discover and retrieve queued service messages for individual clients and <transfer> to get transfer status information for an object.

14.2.1.1 EPP <domain:check> Command

The <idn:check> element, if present, allows to specify the script with the help of the <script> element. It applies to all names submitted. No extension is added to the response of the check command.

If a given name is not suitable for the given script, it is marked as unavailable with the reason of "Invalid".

Using the "attribute mode", the name is reported as "In use", if an identical domain name exists. If the name represents a variant of an existing domain, it is reported as "Blocked", independent of whether the variant actually exists or not.

14.2.1.2 EPP <domain:info> Command

The IDN extension does not provide an element for the info command. However, the response may contain an <idn:infData> element providing additional information.

The current script is reported via the <script> element.

As the registry system uses the "attribute mode", the provided domain name must be the object name. It is not possible to use a variant name, neither existing or non-existing, in which case an "object does not exist" error is generated. The <idn:infData> contains a list of all existing variants that have been defined for the domain.

Note that the extension is omitted if the domain is an ASCII domain name and does not have any variant names.

14.2.1.3 EPP <poll> Command

This extension does not add any element to the EPP <poll> command itself.

14.2.1.4 EPP <transfer> Query Command

This extension does not add any element to the EPP <transfer> query command itself.

14.2.2 EPP Transform Commands

There are five EPP commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes and <update> to change information associated with an object.

14.2.2.1 EPP <domain:create> Command

The create command, which allows the registration of domain objects, or, during the registry sunrise and landrush phases, the application for domain objects, can be augmented by an <idn:create> element in the extension section of the command. This element can carry a language/script identifier as well as a list of variants, as described in the following.

With the help of the <script> element, the used script of the name and variants is defined. The specification of the script is optional. If omitted, an ASCII domain name is assumed.

Along with the domain name itself, up to 10 variants may be specified.

14.2.2.2 EPP <domain:delete> Command

There are no extension elements for the domain delete command and response.

14.2.2.3 EPP <domain:renew> Command

There are no extension elements for the domain renew command and response.

14.2.2.4 EPP <domain:transfer> Command

For the .swiss TLD, the extension element <idn:trnData> is not used. It applies only to instance that use the "object mode" variants model. There is no special behaviour of the transfer command regarding IDNs.

14.2.2.5 EPP <domain:update> Command

For the domain update command, extension elements exist for both the command and response, <idn:update> and <idn:updData> respectively. The latter is only used for variants in the "object mode".

Using the <script> element, the language of an existing domain name and its variants can be changed at a later point in time. This can be useful if it turns out that the domain has been originally registered using a wrong script and a desired variant could not be registered. A precondition for the successful execution of the change of the script is that the domain name the final variants (i.e. after applying any supplied variant additions or deletions within the same extension) are valid names for the new script.

By supplying variant names in the <add> and <rem> section, variants associated with the domains can be added and removed, respectively. The variants added must be valid variants of the domain name. After processing the changes, the limit of 10 variants may not be exceeded.

14.3 Formal Syntax (Schema Definition)

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

<schema targetNamespace="http://xmlns.corenic.net/epp/idn-1.0"
    xmlns:idn="http://xmlns.corenic.net/epp/idn-1.0"
    xmlns:eppcom="urn:ietf:params:xml:ns:eppcom-1.0"
    xmlns="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified"
    attributeFormDefault="unqualified">

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      domain name extension schema for internationalised domain names
      processing.
    </documentation>
  </annotation>

  <import namespace="urn:ietf:params:xml:ns:eppcom-1.0"
    schemaLocation="eppcom-1.0.xsd"/>


  <!-- child elements found in EPP commands -->

  <element name="check" type="idn:checkType"/>
  <element name="create" type="idn:createType"/>
  <element name="update" type="idn:updateType"/>


  <!-- child elements of the <check> command -->

  <complexType name="checkType">
    <sequence>
      <group ref="idn:idnTagGroup"/>
    </sequence>
  </complexType>


  <!-- child elements of the <create> command -->

  <complexType name="createType">
    <sequence>
      <group ref="idn:idnTagGroup" minOccurs="0"/>
      <element name="variants" type="idn:variantListType" minOccurs="0"/>
    </sequence>
  </complexType>


  <!-- child elements of the <update> command -->
  <complexType name="updateType">
    <sequence>
      <element name="add" type="idn:variantListType" minOccurs="0"/>
      <element name="rem" type="idn:variantListType" minOccurs="0"/>
      <element name="chg" minOccurs="0">
        <complexType>
          <sequence>
            <group ref="idn:idnTagGroup" minOccurs="0"/>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>


  <!-- child response elements -->

  <element name="creData" type="idn:bundleDataType"/>
  <element name="updData" type="idn:bundleDataType"/>
  <element name="trnData" type="idn:bundleDataType"/>
  <element name="infData" type="idn:respDataType"/>


  <!-- response elements -->

  <complexType name="respDataType">
    <sequence>
      <group ref="idn:idnTagGroup"/>
      <element name="variants" type="idn:variantListType" minOccurs="0"/>
    </sequence>
  </complexType>


  <!-- bundle information -->

  <complexType name="bundleDataType">
    <sequence>
      <element name="variants" type="idn:variantListType"/>
    </sequence>
  </complexType>


  <!-- common types -->

  <!-- type that allows an empty string only -->

  <simpleType name="emptyTokenType">
    <restriction base="token">
      <length value="0"/>
    </restriction>
  </simpleType>


  <!-- a script code according to ISO 15924, additionally allowing an
       empty string -->

  <simpleType name="scriptType">
    <union memberTypes="idn:emptyTokenType">
      <simpleType>
        <restriction base="token">
          <minLength value="3"/>
          <maxLength value="4"/>
        </restriction>
      </simpleType>
    </union>
  </simpleType>


  <!-- a language code according to RFC 5646, additionally allowing an
       empty string -->

  <simpleType name="languageType">
    <union memberTypes="idn:emptyTokenType language">
    </union>
  </simpleType>


  <!-- elements that represent either a language or a script -->
  <group name="idnTagGroup">
    <choice>
      <element name="lang" type="idn:languageType"/>
      <element name="script" type="idn:scriptType"/>
    </choice>
  </group>


  <!-- a list of variants of a domain name -->

  <complexType name="variantListType">
    <sequence>
      <element name="nameVariant" type="eppcom:labelType"
        minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

</schema>
```

14.4 Examples

In the following examples, "C:" represents lines sent by an EPP client and "S:" represents lines returned by the .swiss Registry EPP server.

14.4.1 EPP <check> Command

14.4.1.1 Example <check> command with script tag:

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <check>
C:      <check xmlns="urn:ietf:params:xml:ns:domain-1.0">
C:        <name>example.swiss</name>
C:      </check>
C:    </check>
C:    <extension>
C:      <check xmlns="http://xmlns.corenic.net/epp/idn-1.0">
C:        <script>Latn</script>
C:      </check>
C:    </extension>
C:    <clTRID>O168899153_1332496264546</clTRID>
C:  </command>
C:</epp>
```

14.4.2 EPP <info> Command

14.4.2.1 Example <info> response with script tag and empty list of variants:

```
S:<?xml version='1.0' encoding='UTF-8'?>
S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S:  <response>
S:    <result code='1000'>
S:      <msg lang='en-US'>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <infData xmlns='urn:ietf:params:xml:ns:domain-1.0'>
S:        <name>example.swiss</name>
S:        <roid>D123456789</roid>
S:        <status s='active'/>
S:        <registrant>abc123</registrant>
S:        <contact type='admin'>def456</contact>
S:        <contacttype='tech'>ghi789</contact>
S:        <ns>
S:          <hostObj>ns1.example.net</hostObj>
S:          <hostObj>ns2.example.net</hostObj>
S:        </ns>
S:        <clID>registrar</clID>
S:        <crID>registrar</crID>
S:        <crDate>2010-09-08T07:06:05.0Z</crDate>
S:        <exDate>2012-09-08T23:59:59.0Z</exDate>
S:        <authInfo>
S:          <pw>secret</pw>
S:        </authInfo>
S:      </infData>
S:    </resData>
S:    <extension>
S:      <infData xmlns="http://xmlns.corenic.net/epp/idn-1.0">
S:        <script>Latn</script>
S:        <variants/>
S:      </infData>
S:    </extension>
S:    <trID>
```

```
S:      <svTRID>ZYX-99958</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

14.4.2.2 Example <info> response with script tag and domain name variants:

```
S:<?xml version='1.0' encoding='UTF-8'?>
S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S:  <response>
S:    <result code='1000'>
S:      <msg lang='en-US'>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <infData xmlns='urn:ietf:params:xml:ns:domain-1.0'>
S:        <name>example.swiss</name>
S:        <roid>D123456789-COM</roid>
S:        <status s='active'/>
S:        <registrant>abc123</registrant>
S:        <contact type='admin'>def456</contact>
S:        <contacttype='tech'>ghi789</contact>
S:        <ns>
S:          <hostObj>ns1.example.net</hostObj>
S:          <hostObj>ns2.example.net</hostObj>
S:        </ns>
S:        <clID>registrar</clID>
S:        <crID>registrar</crID>
S:        <crDate>2010-09-08T07:06:05.0Z</crDate>
S:        <exDate>2012-09-08T23:59:59.0Z</exDate>
S:        <authInfo>
S:          <domain:pw>secret</pw>
S:        </authInfo>
S:      </infData>
S:    </resData>
S:    <extension>
S:      <infData xmlns="http://xmlns.corenic.net/epp/idn-1.0">
S:        <script>Latn</script>
S:        <variants>
S:          <nameVariant>dummy.swiss</nameVariant>
S:          <nameVariant>wrong.swiss</nameVariant>
S:        </variants>
S:      </infData>
S:    </extension>
S:    <trID>
S:      <svTRID>ZYX-99958</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

14.4.3 EPP <create> Command

14.4.3.1 Example <create> command with script tag and empty list of variants:

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <create xmlns="urn:ietf:params:xml:ns:domain-1.0">
C:        <name>example.swiss</name>
C:        <period unit="y">1</period>
C:        <ns>
C:          <hostObj>ns1.example.net</hostObj>
C:          <hostObj>ns2.example.net</hostObj>
C:        </ns>
C:        <registrant>abc123</registrant>
C:        <contact type="admin">def456</contact>
C:        <contact type="tech">ghi789</contact>
C:        <authInfo>
C:          <pw>secret42</pw>
C:        </authInfo>
C:      </create>
C:    </create>
C:    <extension>
C:      <create xmlns="http://xmlns.corenic.net/epp/idn-1.0">
C:        <script>Latn</script>
C:      </create>
C:    </extension>
C:    <clTRID>abc-00042</clTRID>
C:  </command>
C:</epp>
```

14.4.3.2 Example <create> command with script tag and domain name variants:

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <create xmlns="urn:ietf:params:xml:ns:domain-1.0">
C:        <name>example.swiss</name>
C:        <period unit="y">1</period>
C:        <ns>
C:          <hostObj>ns1.example.net</hostObj>
C:          <hostObj>ns2.example.net</hostObj>
C:        </ns>
C:        <registrant>abc123</registrant>
C:        <contact type="admin">def456</contact>
C:        <contact type="tech">ghi789</contact>
C:        <authInfo>
C:          <pw>secret42</pw>
C:        </authInfo>
C:      </create>
C:    </create>
C:    <extension>
C:      <create xmlns="http://xmlns.corenic.net/epp/idn-1.0">
C:        <script>Latn</script>
C:        <variants>
C:          <nameVariant>dummy.swiss</nameVariant>
C:          <nameVariant>wrong.swiss</nameVariant>
C:        </variants>
C:      </create>
C:    </extension>
C:    <clTRID>abc-00042</clTRID>
C:  </command>
C:</epp>
```

14.4.4 EPP <update> Command

14.4.4.1 Example <update> command changing the script of a domain:

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:domain-1.0">
C:        <name>example.swiss</name>
C:        <chg/>
C:      </update>
C:    </update>
C:    <extension>
C:      <update xmlns="http://xmlns.corenic.net/epp/idn-1.0">
C:        <chg>
C:          <script>Latn</script>
C:        </chg>
C:      </update>
C:    </extension>
C:    <clTRID>abc-00042</clTRID>
C:  </command>
C:</epp>
```

14.4.4.2 Example <update> command adding and removing domain name variants:

```
C:<?xml version="1.0" encoding="UTF-8"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:domain-1.0">
C:        <name>example.swiss</name>
C:        <chg/>
C:      </update>
C:    </update>
C:    <extension>
C:      <update xmlns="http://xmlns.corenic.net/epp/idn-1.0">
C:        <add>
C:          <nameVariant>silly.swiss</nameVariant>
C:          <nameVariant>right.swiss</nameVariant>
C:        </add>
C:        <rem>
C:          <nameVariant>dummy.swiss</nameVariant>
C:          <nameVariant>wrong.swiss</nameVariant>
C:        </rem>
```

```
C:      </update>
C:    </extension>
C:    <clTRID>abc-00042</clTRID>
C:  </command>
C:</epp>
```

15. Extension for Contact Eligibility (Version 1.0)

The CORE Registration System used to operate the .swiss TLD provides a proprietary EPP extension for providing contact eligibility information. This document describes Version 1.0 of that EPP extension.

Note that a new version of this extension (1.1) is available that superseded this version and should be implemented instead if it is supported by the EPP system in question (even if this version 1.0 is offered as a legacy extension).

15.1 Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) described in RFC 5730. This mapping is an extension of the contact mapping described in RFC 5733. It is specified using the Extensible Markup Language (XML) and XML Schema notation.

This extension serves the purpose of specifying contact eligibility information, i.e. additional information to be associated with a contact object that allows the registry to determine a contact's (usually the registrant's) eligibility for acquiring or keeping a certain domain name it is associated with. Depending on a registry's policies, the specification of certain contact eligibility information may be be mandatory to obtain or retain a specific domain name. For example, a registry may mandate that a domain's registrant is a registered enterprise, and may thus require a registrar to specify the company's enterprise ID as contact eligibility information for contacts used as a registrant for the company's domains.

In its version 1.0, the extension allows the specification of an ASCII-only "enterprise ID" string representing the registered enterprise ID of the company represented by the contact. The information is inspected by the registry to decide whether the company is eligible to obtain a certain domain name according to registry policy.

A contact's enterprise ID may be specified when creating the contact with the <contact:create> EPP command. This information may be changed or removed later using the <contact:update> EPP command. The current enterprise ID on file for a contact may be inquired using the <contact:info> command. In addition, if a contact carries eligibility information, the response to a <contact:info> command will provide information about the contact's validation status.

Depending on the policy of a registry supporting this extension, it may be optional or mandatory to provide eligibility information with a contact object. If eligibility information is optional, the registry policy may prevent registrars from using contacts lacking eligibility information from being used in certain roles within a domain.

15.2 EPP Command Mapping

This section deals with the specific command mappings for the .swiss TLD EPP extension for contact eligibility information.

In the following, the respective root elements of the extensions are mentioned. If used, they must be placed or expected within the optional <extension> element at the proper location in the XML document representing the EPP command or response, as described in RFC 5730. Note that the use of the "el:" XML namespace prefix is for documentation purposes only. Conforming to the "Namespaces in XML 1.1" standard, EPP and the registry implementation take only the associated namespace URI into account, and not the prefix itself. So actually any prefix or even the default namespace may be used in requests and must be expected in responses.

The eligibility extension is only used in relation to contact objects. It will not occur in commands that are related to domain and host objects.

15.2.1 EPP Query Commands

There are four EPP commands to retrieve object information: <check> to find out whether an object is known to the server, <info> to ask for detailed information associated with an object, <poll> to discover and retrieve queued service messages for individual clients and <transfer> to get transfer status information for an object.

15.2.1.1 EPP <contact:check> Command

This extension does not add any element to the EPP <check> command.

15.2.1.2 EPP <contact:info> Command

The eligibility extension does not provide an element for the info command. Additional elements are defined for the <info> response. When an <info> command has been processed successfully for a contact carrying eligibility information, the EPP <extension> element in the response contains a child <el:infData> element that identifies the contact eligibility extension namespace. The <el:infData> element contains a <el:enterpriseID> element, which contains the enterprise ID string currently present for the inquired contact. Furthermore, the <el:infData> element also contains an <el:validationStatus> element denoting the contact's validation status, represented by one of these three values:

- **pendingValidation**: a validation of the contact's eligibility data is pending
- **validated**: the contact's eligibility data was successfully validated, i.e. it was found to be valid; the contact may thus be used in successful domain registrations
- **invalid**: the contact's eligibility data failed the validation, i.e. it was found to be invalid; the contact is unsuitable for use in successful domain registrations

An example of an <info> response containing the extension can be found in the "Examples" section below.

15.2.1.3 EPP <poll> Command

This extension does not add any element to the EPP <poll> command.

15.2.1.4 EPP <transfer> Query Command

This extension does not add any element to the EPP <transfer> query command.

15.2.2 EPP Transform Commands

There are four EPP commands to transform contact objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <transfer> to manage object sponsorship changes and <update> to change information associated with an object.

15.2.2.1 EPP <contact:create> Command

The create command, which allows the registration of contact objects, can be augmented by a <el:create> element in the extension section of the command that identifies the contact eligibility extension namespace. If specified, the <el:create> element must contain a <el:enterpriseID> element, which content specifies the contact's enterprise ID. The enterprise ID is a non-empty, ASCII-only string with a maximum length of 30 characters.

After creation, the contact's validation status will be "pendingValidation". It may later change to "validated" or "invalid", depending on the outcome of the registry's validation. Note that the validation process for a contact may not be started until the contact is used in a domain or application, as the registry may choose to only conduct the validation if it is required for deciding whether or not to allocate a domain. A contact's validation status may thus remain "pendingValidation" indefinitely until it is associated with a domain or application.

An example of a <create> command using the extension can be found in the "Examples" section below.

15.2.2.2 EPP <contact:delete> Command

There are no extension elements for the contact delete command and response.

15.2.2.3 EPP <contact:transfer> Command

There are no extension elements for the transfer command and response.

15.2.2.4 EPP <contact:update> Command

For the contact update command, extension elements exist for the command, allowing to change or remove the enterprise ID for an existing contact. This is done by using an <el:update> element in the extension section of the command that identifies the contact eligibility extension namespace. The <el:update> element must contain an <el:chg> element, which in turn must contain an <el:enterpriseID> element, which content specifies the contacts's new enterprise ID. To remove the enterprise ID from a contact entirely, the <el:enterpriseID> element may be empty in the case of updates.

Note that any change to a contact's eligibility information may trigger re-validation by the registry and may thus change the contact's validation status, plus potentially the status of domains or applications associated with the contact.

15.3 Formal Syntax (Schema Definition)

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://xmlns.corenic.net/epp/contact-eligibility-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.0"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      extension schema for provisioning registration eligibility information
      for contact objects
    </documentation>
  </annotation>


  <!-- child elements found in EPP commands -->

  <element name="create" type="el:createType"/>
  <element name="update" type="el:updateType"/>

  <!-- child elements found in EPP responses -->

  <element name="infData" type="el:infDataType"/>


  <!-- optional enterprise ID type for update commands (allows removing it) -->

  <simpleType name="optEnterpriseIDType">
    <restriction base="string">
      <maxLength value="30"/>
      <pattern value="\p{IsBasicLatin}*"/>
    </restriction>
  </simpleType>


  <!-- non-empty enterprise ID type for create commands and info responses -->

  <simpleType name="enterpriseIDType">
    <restriction base="string">
      <minLength value="1"/>
      <maxLength value="30"/>
      <pattern value="\p{IsBasicLatin}*"/>
    </restriction>
  </simpleType>


  <!-- possible status values for a contact validation -->

  <simpleType name="validationStatusType">
    <restriction base="token">
      <enumeration value="pendingValidation"/>
      <enumeration value="validated"/>
      <enumeration value="invalid"/>
    </restriction>
  </simpleType>


  <!-- create extension -->
  <complexType name="createType">
    <sequence>
      <element name="enterpriseID" type="el:enterpriseIDType"/>
    </sequence>
  </complexType>


  <!-- update extension -->

  <complexType name="updateType">
    <sequence>
      <element name="chg" type="el:chgType"/>
    </sequence>
  </complexType>

  <complexType name="chgType">
    <sequence>
      <!-- new value for the enterprise ID; may be left empty to remove it -->
      <element name="enterpriseID" type="el:optEnterpriseIDType"/>
    </sequence>
  </complexType>


  <!-- info extension -->

  <complexType name="infDataType">
    <sequence>
      <element name="enterpriseID" type="el:enterpriseIDType"/>
      <element name="validationStatus" type="el:validationStatusType"/>
    </sequence>
  </complexType>

</schema>
```

15.4 Examples

In the following examples, "C:" represents lines sent by an EPP client and "S:" represents lines returned by the .swiss Registry EPP server.

15.4.1 EPP <info> Command

15.4.1.1 Example <info> response with eligibility information:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <infData xmlns="urn:ietf:params:xml:ns:contact-1.0">
S:        <id>sh8013</id>
S:        <roid>SH8013-REP</roid>
S:        <status s="linked"/>
S:        <status s="clientDeleteProhibited"/>
S:        <postalInfo type="int">
S:          <name>Hans Mustermann</name>
S:          <org>Musterfirma</org>
S:          <addr>
S:            <street>Beispielweg 1</street>
S:            <city>Bern</city>
S:            <sp>Bern</sp>
S:            <pc>123456</pc>
S:            <cc>CH</cc>
S:          </addr>
S:        </postalInfo>
S:        <voice x="1234">+1.7035555555</voice>
S:        <fax>+1.7035555556</fax>
S:        <email>jdoe@example.com</email>
S:        <clID>ClientY</clID>
S:        <crID>ClientX</crID>
S:        <crDate>1999-04-03T22:00:00.0Z</crDate>
S:        <upID>ClientX</upID>
S:        <upDate>1999-12-03T09:00:00.0Z</upDate>
S:        <trDate>2000-04-08T09:00:00.0Z</trDate>
S:        <authInfo>
S:          <pw>2fooBAR</pw>
```

```
S:        </authInfo>
S:        <disclose flag="0">
S:          <voice/>
S:          <email/>
S:        </disclose>
S:      </infData>
S:    </resData>
S:    <extension>
S:      <el:infData xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.0">
S:        <el:enterpriseID>DE-ID-122322322</el:enterpriseID>
S:        <el:validationStatus>validationPending</el:validationStatus>
S:      </el:infData>
S:    </extension>
S:    <trID>
S:      <svTRID>ZYX-99958</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

15.4.2 EPP <create> Command

15.4.2.1 Example <create> command specifying eligibility information:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <create xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:        <postalInfo type="int">
C:          <name>Hans Mustermann</name>
C:          <org>Musterfirma</org>
C:          <addr>
C:            <street>Beispielweg 1</street>
C:            <city>Bern</city>
C:            <sp>Bern</sp>
C:            <pc>123456</pc>
C:            <cc>CH</cc>
C:          </addr>
C:        </postalInfo>
C:        <voice x="1234">+1.7035555555</voice>
C:        <fax>+1.7035555556</fax>
C:        <email>jdoe@example.com</email>
C:        <authInfo>
C:          <pw>2fooBAR</pw>
C:        </authInfo>
C:        <disclose flag="0">
C:          <voice/>
C:          <email/>
C:        </disclose>
C:      </create>
C:    </create>
C:    <extension>
C:      <el:create xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.0">
C:        <el:enterpriseID>DE-ID-122322322</el:enterpriseID>
C:      </el:create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

15.4.3 EPP <update> Command

15.4.3.1 Example <update> command changing the eligibility information of a contact, along with other contact data:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:        <add>
C:          <status s="clientDeleteProhibited"/>
C:        </add>
C:        <chg>
```

```
C:          <postalInfo type="int">
C:            <name>Hans Mustermann</name>
C:            <org/>
C:            <addr>
C:              <street>Beispielweg 1</street>
C:              <city>Bern</city>
C:              <sp>Bern</sp>
C:              <pc>123456</pc>
C:              <cc>CH</cc>
C:            </addr>
C:          </postalInfo>
C:          <voice>+1.7034444444</voice>
C:          <fax/>
C:          <authInfo>
C:            <pw>2fooBAR</pw>
C:          </authInfo>
C:          <disclose flag="1">
C:            <voice/>
C:            <email/>
C:          </disclose>
C:        </chg>
C:      </update>
C:    </update>
C:    <extension>
C:      <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.0">
C:        <el:chg>
C:          <el:enterpriseID>DE-ID-122322325</el:enterpriseID>
C:        </el:chg>
C:      </el:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

15.4.3.2 Example <update> command changing the eligibility information of a contact, retaining other contact data:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:      </update>
C:    </update>
C:    <extension>
C:      <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.0">
C:        <el:chg>
C:          <el:enterpriseID>DE-ID-122322324</el:enterpriseID>
C:        </el:chg>
C:      </el:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

15.4.3.3 Example <update> command removing the eligibility information of a contact, retaining other contact data:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:      </update>
C:    </update>
C:    <extension>
C:      <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.0">
C:        <el:chg>
C:          <el:enterpriseID/>
C:        </el:chg>
C:      </el:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

16. Extension for Contact Eligibility (Version 1.1)

The CORE Registration System used to operate the .swiss TLD provides a proprietary EPP extension for providing contact eligibility information. This document describes Version 1.1 of that EPP extension.

16.1 Introduction

This document describes an extension mapping for version 1.0 of the Extensible Provisioning Protocol (EPP) described in RFC 5730. This mapping is an extension of the contact mapping described in RFC 5733. It is specified using the Extensible Markup Language (XML) and XML Schema notation.

This extension serves the purpose of specifying contact eligibility information, i.e. additional information to be associated with a contact object that allows the registry to determine a contact's (usually the registrant's) eligibility for acquiring or keeping a certain domain name it is associated with. Depending on a registry's policies, the specification of certain contact eligibility information may be be mandatory to obtain or retain a specific domain name. For example, a registry may mandate that a domain's registrant is a registered enterprise, and may thus require a registrar to specify the company's company/enterprise ID as contact eligibility information for contacts used as a registrant for the company's domains. Likewise, if a domain's registrant must be a citizen of a certain country or region, the registry may require a registrar to specify the registrant's person ID (as issued by the country or region) as contact eligibility information.

16.1.1 Typed Public IDs

In its version 1.1, the extension allows the specification of one or more ASCII-only **public IDs** associated with the company or person represented by the contact. Each public ID string has a mandatory type, represented by a "type" string whose permissible content is defined by registry policy.

16.1.2 Natural Person Flag

In addition to the public IDs, the extension also allows to specify whether or not the contact represents a **natural person** or not.

16.1.3 Server Policy

The information is inspected by the registry to decide whether the company is eligible to obtain a certain domain name according to registry policy.

The public ID(s) and "natural person" flag may be specified when creating the contact with the <contact:create> EPP command. This information may be changed or removed later using the <contact:update> EPP command. The current public ID(s) and "natural person" flag on file for a contact may be inquired using the <contact:info> command. In addition, if a contact carries any eligibility information, the response to a <contact:info> command will provide information about the contact's validation status.

Depending on the policy of a registry supporting this extension, it may be optional or mandatory to provide eligibility information with a contact object. If eligibility information is optional, the registry policy may prevent registrars from using contacts lacking eligibility information from being used in certain roles within a domain. If a contact has eligibility information and is associated with an active domain in a certain role, the registry policy may prevent the removal of eligibility information from said contact. Registry policy may also restrict the number of specifiable public IDs, e.g. some registries may only allow one public ID per contact.

16.2 EPP Command Mapping

This section deals with the specific command mappings for the .swiss TLD EPP extension for contact eligibility information.

In the following, the respective root elements of the extensions are mentioned. If used, they must be placed or expected within the optional <extension> element at the proper location in the XML document representing the EPP command or response, as described in RFC 5730. Note that the use of the "el:" XML namespace prefix is for documentation purposes only. Conforming to the "Namespaces in XML 1.1" standard, EPP and the registry implementation take only the associated namespace URI into account, and not the prefix itself. So actually any prefix or even the default namespace may be used in requests and must be expected in responses.

The eligibility extension is only used in relation to contact objects. It will not occur in commands that are related to domain and host objects.

16.2.1 EPP Query Commands

There are four EPP commands to retrieve object information: <check> to find out whether an object is known to the server, <info> to ask for detailed information associated with an object, <poll> to discover and retrieve queued service messages for individual clients and <transfer> to get transfer status information for an object.

16.2.1.1 EPP <contact:check> Command

This extension does not add any element to the EPP <check> command.

16.2.1.2 EPP <contact:info> Command

The eligibility extension does not provide an element for the info command. Additional elements are defined for the <info> response, though. When an <info> command has been processed successfully for a contact carrying eligibility information, the EPP <extension> element in the response contains a child <el:infData> element that identifies the contact eligibility extension namespace.

The <el:infData> element may contain one or more <el:publicId> elements, which contain the public ID strings currently present for the inquired contact, each with their type contained in a "type" XML attribute.

Also, the <el:infData> element may contain an <el:naturalPerson> element, which contains the contact's "natural person" status as a boolean value ("true" denoting "natural person", "false" denoting "no natural person").

Furthermore, the <el:infData> element may contain an <el:validationStatus> element denoting the contact's validation status, represented by one of these three values:

- **pendingValidation**: a validation of the contact's eligibility data is pending
- **validated**: the contact's eligibility data was successfully validated, i.e. it was found to be valid; the contact may thus be used in successful domain registrations
- **invalid**: the contact's eligibility data failed the validation, i.e. it was found to be invalid; the contact is unsuitable for use in successful domain registrations

An example of an <info> response containing the extension can be found in the "Examples" section below.

16.2.1.3 EPP <poll> Command

This extension does not add any element to the EPP <poll> command.

16.2.1.4 EPP <transfer> Query Command

This extension does not add any element to the EPP <transfer> query command.

16.2.2 EPP Transform Commands

There are four EPP commands to transform contact objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <transfer> to manage object sponsorship changes and <update> to change information associated with an object.

16.2.2.1 EPP <contact:create> Command

The create command, which allows the registration of contact objects, can be augmented by a <el:create> element in the extension section of the command that identifies the contact eligibility extension namespace. If specified, the <el:create> element may contain an <el:publicId> element whose content specifies one of the contact's public IDs, each along with the its type contained in the mandatory "type" attribute. A public ID is a non-empty, ASCII-only string with a maximum length of 30 characters.

Also, the <el:create> element may contain an <el:naturalPerson> element, which must contain the contact's "natural person" status as a boolean value ("true" denoting "natural person", "false" denoting "no natural person").

After creation, the contact's validation status will be "pendingValidation". It may later change to "validated" or "invalid", depending on the outcome of the registry's validation. Note that the validation process for a contact may not be started until the contact is used in a domain or application, as the registry may choose to only conduct the validation if it is required for deciding whether or not to allocate a domain. A contact's validation status may thus remain "pendingValidation" indefinitely until it is associated with a domain or application.

An example of a <create> command using the extension can be found in the "Examples" section below.

16.2.2.2 EPP <contact:delete> Command

There are no extension elements for the contact delete command and response.

16.2.2.3 EPP <contact:transfer> Command

There are no extension elements for the transfer command and response.

16.2.2.4 EPP <contact:update> Command

For the contact update command, extension elements exist for the command, allowing to change or remove the public IDs and/or "natural person" flag for an existing contact. This is done by using an <el:update> element in the extension section of the command that identifies the contact eligibility extension namespace. The <el:update> element must contain an <el:chg> element, which in turn may contain one or more <el:publicId> elements and/or an <el:naturalPerson/> element to remove or update the respective data. To remove the public ID of a type from a contact entirely, the <el:publicId> element may be left empty, with only the "type" attribute being present to signal the removal of the public ID of that type. Likewise, to remove the "natural person" flag from a contact entirely, the <el:naturalPerson> element may be left empty to signal the removal.

Note that any change to a contact's eligibility information may trigger re-validation by the registry and may thus change the contact's validation status, plus potentially the status of domains or applications associated with the contact.

16.3 Formal Syntax (Schema Definition)

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://@XMLNS_DOMAIN@/epp/contact-eligibility-1.1"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1"
  elementFormDefault="qualified">

  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      extension schema for provisioning registration eligibility information
      for contact objects
    </documentation>
  </annotation>


  <!-- child elements found in EPP commands -->

  <element name="create" type="el:createType"/>
  <element name="update" type="el:updateType"/>

  <!-- child elements found in EPP responses -->

  <element name="infData" type="el:infDataType"/>


  <!-- optional string type for update commands (allows removing items) -->

  <simpleType name="optLatinType">
    <restriction base="string">
      <pattern value="\p{IsBasicLatin}*"/>
    </restriction>
  </simpleType>
  <!-- non-empty string type for create commands and info responses -->

  <simpleType name="reqLatinType">
    <restriction base="string">
      <minLength value="1"/>
      <pattern value="\p{IsBasicLatin}*"/>
    </restriction>
  </simpleType>


  <!-- possible status values for a contact validation -->

  <simpleType name="validationStatusType">
    <restriction base="token">
      <enumeration value="pendingValidation"/>
      <enumeration value="validated"/>
      <enumeration value="invalid"/>
    </restriction>
  </simpleType>


  <!-- possible values for the "natural person" flag -->

  <simpleType name="naturalPersonType">
    <restriction base="token">
      <enumeration value="true"/>
      <enumeration value="false"/>
      <enumeration value=""/> <!-- empty value means "unspecified" -->
    </restriction>
  </simpleType>


  <!-- required public ID + mandatory type attribute -->

  <complexType name="reqPublicId">
    <simpleContent>
      <extension base="el:reqLatinType">
        <attribute name="type" use="required"/>
      </extension>
    </simpleContent>
  </complexType>


  <!-- optional public ID + mandatory type attribute -->

  <complexType name="optPublicId">
    <simpleContent>
      <extension base="el:optLatinType">
        <attribute name="type" use="required"/>
      </extension>
    </simpleContent>
  </complexType>


  <!-- create extension -->

  <complexType name="createType">
    <sequence>
      <!-- multiple public IDs may be specified, but each type only once -->
      <element name="publicId" type="el:reqPublicId" minOccurs="0"
               maxOccurs="unbounded"/>

      <element name="naturalPerson" type="el:naturalPersonType" minOccurs="0"/>
    </sequence>
  </complexType>


  <!-- update extension -->

  <complexType name="updateType">
    <sequence>
      <element name="chg" type="el:chgType"/>
    </sequence>
  </complexType>


  <!-- type for changing / removing eligibility data -->
  <complexType name="chgType">
    <sequence>
      <!-- multiple public IDs may be specified, but each type only once;
           empty elements (with mandatory type) indicate that the public ID
           for that type shall be removed -->
      <element name="publicId" type="el:optPublicId" minOccurs="0"
               maxOccurs="unbounded"/>

      <!-- empty element asks the natural person flag to be cleared -->
      <element name="naturalPerson" type="el:naturalPersonType" minOccurs="0"/>
    </sequence>
  </complexType>


  <!-- info extension -->

  <complexType name="infDataType">
    <sequence>
      <!-- multiple public IDs may occur, but each type only once -->
      <element name="publicId" type="el:reqPublicId" minOccurs="0"
               maxOccurs="unbounded"/>

      <element name="naturalPerson" type="el:naturalPersonType" minOccurs="0"/>

      <element name="validationStatus" type="el:validationStatusType"
               minOccurs="0"/>
    </sequence>
  </complexType>
</schema>
```

16.4 Examples

In the following examples, "C:" represents lines sent by an EPP client and "S:" represents lines returned by the .swiss Registry EPP server.

16.4.1 EPP <info> Command

16.4.1.1 Example <info> response with eligibility information:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <infData xmlns="urn:ietf:params:xml:ns:contact-1.0">
S:        <id>sh8013</id>
S:        <roid>SH8013-REP</roid>
S:        <status s="linked"/>
S:        <status s="clientDeleteProhibited"/>
S:        <postalInfo type="int">
S:          <name>Hans Mustermann</name>
S:          <org>Musterfirma</org>
S:          <addr>
S:            <street>Beispielweg 1</street>
S:            <city>Bern</city>
S:            <sp>Bern</sp>
S:            <pc>123456</pc>
S:            <cc>CH</cc>
S:          </addr>
S:        </postalInfo>
S:        <voice x="1234">+1.7035555555</voice>
S:        <fax>+1.7035555556</fax>
S:        <email>jdoe@example.com</email>
S:        <clID>ClientY</clID>
S:        <crID>ClientX</crID>
S:        <crDate>1999-04-03T22:00:00.0Z</crDate>
S:        <upID>ClientX</upID>
S:        <upDate>1999-12-03T09:00:00.0Z</upDate>
S:        <trDate>2000-04-08T09:00:00.0Z</trDate>
S:        <authInfo>
S:          <pw>2fooBAR</pw>
S:        </authInfo>
S:        <disclose flag="0">
S:          <voice/>
S:          <email/>
```

```
S:        </disclose>
S:      </infData>
S:    </resData>
S:    <extension>
S:      <el:infData xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
S:        <el:publicId type="ch-upi">756.1234.5678.90</el:publicId>
S:        <el:naturalPerson>true</el:naturalPerson>
S:        <el:validationStatus>validationPending</el:validationStatus>
S:      </el:infData>
S:    </extension>
S:    <trID>
S:      <svTRID>ZYX-99958</svTRID>
S:    </trID>
S:  </response>
S:</epp>
```

16.4.2 EPP <create> Command

16.4.2.1 Example <create> command specifying eligibility information:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <create>
C:      <create xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:        <postalInfo type="int">
C:          <name>Hans Mustermann</name>
C:          <org>Musterfirma</org>
C:          <addr>
C:            <street>Beispielweg 1</street>
C:            <city>Bern</city>
C:            <sp>Bern</sp>
C:            <pc>123456</pc>
C:            <cc>CH</cc>
C:          </addr>
C:        </postalInfo>
C:        <voice x="1234">+1.7035555555</voice>
C:        <fax>+1.7035555556</fax>
C:        <email>jdoe@example.com</email>
C:        <authInfo>
C:          <pw>2fooBAR</pw>
C:        </authInfo>
C:        <disclose flag="0">
C:          <voice/>
C:          <email/>
C:        </disclose>
C:      </create>
C:    </create>
C:    <extension>
C:      <el:create xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
C:        <el:publicId type="ch-upi">756.1234.5678.90</el:publicId>
C:        <el:naturalPerson>true</el:naturalPerson>
C:      </el:create>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

16.4.3 EPP <update> Command

16.4.3.1 Example <update> command changing the eligibility information (public ID of type "ch-uid" and the natural person flag) of a contact, along with other contact data:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:        <add>
C:          <status s="clientDeleteProhibited"/>
C:        </add>
C:        <chg>
C:          <postalInfo type="int">
```

```
C:              <name>Hans Mustermann</name>
C:              <org/>
C:              <addr>
C:                <street>Beispielweg 1</street>
C:                <city>Bern</city>
C:                <sp>Bern</sp>
C:                <pc>123456</pc>
C:                <cc>CH</cc>
C:              </addr>
C:            </postalInfo>
C:            <voice>+1.7034444444</voice>
C:            <fax/>
C:            <authInfo>
C:              <pw>2fooBAR</pw>
C:            </authInfo>
C:            <disclose flag="1">
C:              <voice/>
C:              <email/>
C:            </disclose>
C:          </chg>
C:        </update>
C:      </update>
C:      <extension>
C:        <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
C:          <el:chg>
C:            <el:publicId type="ch-uid">CHE-123.456.789</el:publicId>
C:            <el:naturalPerson>false</el:naturalPerson>
C:          </el:chg>
C:        </el:update>
C:      </extension>
C:      <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

16.4.3.2 Example <update> command changing only the eligibility information of a contact, retaining other contact data:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:    <command>
C:      <update>
C:        <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:          <id>sh8013</id>
C:        </update>
C:      </update>
C:      <extension>
C:        <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
C:          <el:chg>
C:            <el:publicId type="ch-uid">CHE-123.456.789</el:publicId>
C:            <el:naturalPerson>false</el:naturalPerson>
C:          </el:chg>
C:        </el:update>
C:      </extension>
C:      <clTRID>ABC-12345</clTRID>
C:    </command>
C:</epp>
```

16.4.3.3 Example <update> command removing the "public ID" eligibility information of a contact, retaining other contact data and retaining the "natural person" flag:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:    <command>
C:      <update>
C:        <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:          <id>sh8013</id>
C:        </update>
C:      </update>
C:      <extension>
C:        <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
C:          <el:chg>
C:            <el:publicId type="ch-uid"/>
C:          </el:chg>
C:        </el:update>
C:      </extension>
```

```
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

16.4.3.4 Example <update> command removing the "natural person" eligibility information of a contact, retaining other contact data and the public IDs:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:      </update>
C:    </update>
C:    <extension>
C:      <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
C:        <el:chg>
C:          <el:naturalPerson/>
C:        </el:chg>
C:      </el:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

16.4.3.5 Example <update> command replacing the "public ID" eligibility information of a contact of another type, retaining other contact data and retaining the "natural person" flag:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:      </update>
C:    </update>
C:    <extension>
C:      <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
C:        <el:chg>
C:          <el:publicId type="ch-uid"/>
C:          <el:publicId type="gleif-lei">123400ABCDEFHIJKLM99</el:publicId>
C:        </el:chg>
C:      </el:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

16.4.3.6 Example <update> command removing all the eligibility information of a contact, retaining other contact data:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C:  <command>
C:    <update>
C:      <update xmlns="urn:ietf:params:xml:ns:contact-1.0">
C:        <id>sh8013</id>
C:      </update>
C:    </update>
C:    <extension>
C:      <el:update xmlns:el="http://xmlns.corenic.net/epp/contact-eligibility-1.1">
C:        <el:chg>
C:          <el:publicId type="gleif-lei"/>
C:          <el:naturalPerson/>
C:        </el:chg>
C:      </el:update>
C:    </extension>
C:    <clTRID>ABC-12345</clTRID>
C:  </command>
C:</epp>
```

17. IDN Table for the Latin Script

TLD: .swiss

Script Tag: Latn

Script Description: Latin

Version: 1.0

Effective Date: 2012-04-04

Registry: .swiss Registry

The .swiss Latin script IDN implementation provides a feature that mitigates homographic attacks by blocking a name from being registered if a similar name is already registered.

For every registered Latin script IDN a canonical string will be computed and stored in the SRS database. The canonical string is produced by mapping every character in the IDN U-label to the corresponding canonical character as specified in the character table below. Once a Latin script IDN is registered, other domain names that map to the same canonical string will be unavailable for registration.

The table below lists the characters allowed in the Unicode representation of IDNs associated with the Latin script. The second column shows the canonical code point corresponding to the first column. The third column lists the other variants that also map to the same canonical character in the second column. Columns are delimited by semicolons. The "#" symbol denotes the start of a comment that continues to the end of line.

The table is based upon the list of available code points from the .eu Latin IDN table. Due to changes mentioned in IDNA2008 a few adjustments have been made. The German character U+00DF (ß) has been added to the list and the Latin middle dot (U+00B7) has replaced the Latin small letter L with middle dot (U+0140), which will always be split to the Latin small letter L (U+006C) plus the Latin middle dot (U+00B7) before the Punycode algorithm is applied. For this middle dot, the following special rules apply (the same rules exist for the Catalan .cat registry): The “ela geminada”, represented in Unicode as U+013F U+004C ĿL (up-per case) resp. U+0140 U+006C ŀl (lower case), needs to be transliterated (e.g., by the NamePrep process) into the character sequence U+006C U+00B7 U+006C (l·l) which appears in the list below. Therefore, it can be used in domain names. The character U+00B7 · (middle dot) cannot be used individually, but only in combination with both preceding and following letter L.

The table presented here is in compliance with the ICANN Guidelines for the Implementation of Internationalized Domain Names Version 3.0 and is intended for publication in the IANA Repository of TLD IDN Practices, for the information of prospective holders of domains in .swiss and for the users of resources within those domains.

```
######
###### Hyphen
  U+002D;U+002D;U+00B7  ##  HYPHEN-MINUS
######
###### Digits
  U+0030;U+0030  ##  DIGIT ZERO
  U+0031;U+0031  ##  DIGIT ONE
  U+0032;U+0032  ##  DIGIT TWO
  U+0033;U+0033  ##  DIGIT THREE
  U+0034;U+0034  ##  DIGIT FOUR
  U+0035;U+0035  ##  DIGIT FIVE
  U+0036;U+0036  ##  DIGIT SIX
  U+0037;U+0037  ##  DIGIT SEVEN
  U+0038;U+0038  ##  DIGIT EIGHT
  U+0039;U+0039  ##  DIGIT NINE
######
###### Basic Latin
  U+0061;U+0061;U+00E0,U+00E1,U+00E2,U+00E3,U+00E4,U+00E5,U+0101,U+0103,U+0105  ##  a LATIN SMALL LETTER A
  U+0062;U+0062  ##  b LATIN SMALL LETTER B
  U+0063;U+0063;U+00E7,U+0107,U+0109,U+010B,U+010D ##  c LATIN SMALL LETTER C
  U+0064;U+0064;U+00F0,U+00FE,U+010F,U+0111  ##  d LATIN SMALL LETTER D
  U+0065;U+0065;U+00E8,U+00E9,U+00EA,U+00EB,U+0113,U+0115,U+0117,U+0119,U+011B  ##  e LATIN SMALL LETTER E
  U+0066;U+0066  ##  f LATIN SMALL LETTER F
  U+0067;U+0067;U+011D,U+011F,U+0121,U+0123  ##  g LATIN SMALL LETTER G
  U+0068;U+0068;U+0125,U+0127  ##  h LATIN SMALL LETTER H
  U+0069;U+0069;U+00EC,U+00ED,U+00EE,U+00EF,U+0129,U+012B,U+012D,U+012F,U+0131  ##  i LATIN SMALL LETTER I
  U+006A;U+006A;U+0135  ##  j LATIN SMALL LETTER J
  U+006B;U+006B;U+0137  ##  k LATIN SMALL LETTER K
  U+006C;U+006C;U+013A,U+013C,U+013E,U+0142 ##  l LATIN SMALL LETTER L
  U+006D;U+006D  ##  m LATIN SMALL LETTER M
  U+006E;U+006E;U+00F1,U+0144,U+0146,U+0148,U+014B  ##  n LATIN SMALL LETTER N
  U+006F;U+006F;U+00F2,U+00F3,U+00F4,U+00F5,U+00F6,U+00F8,U+014D,U+014F,U+0151  ##  o LATIN SMALL LETTER O
  U+0070;U+0070  ##  p LATIN SMALL LETTER P
  U+0071;U+0071  ##  q LATIN SMALL LETTER Q
  U+0072;U+0072;U+0155,U+0157,U+0159  ##  r LATIN SMALL LETTER R
  U+0073;U+0073;U+015B,U+015D,U+0161,U+0219  ##  s LATIN SMALL LETTER S
  U+0074;U+0074;U+0165,U+0167,U+021B ##  t LATIN SMALL LETTER T
  U+0075;U+0075;U+00F9,U+00FA,U+00FB,U+00FC,U+0169,U+016B,U+016D,U+016F,U+0171,U+0173  ##  u LATIN SMALL LETTER U
  U+0076;U+0076  ##  v LATIN SMALL LETTER V
  U+0077;U+0077;U+0175  ##  w LATIN SMALL LETTER W
  U+0078;U+0078  ##  x LATIN SMALL LETTER X
  U+0079;U+0079;U+00FD,U+00FF,U+0177  ##  y LATIN SMALL LETTER Y
  U+007A;U+007A;U+017A,U+017C,U+017E  ##  z LATIN SMALL LETTER Z
######
###### Latin-1 supplement
  U+00B7;U+002D;U+002D  ##  · LATIN MIDDLE DOT
  U+00DF;U+0073 U+0073  ##  ß LATIN SMALL LETTER SHARP S
  U+00E0;U+0061;U+0061,U+00E1,U+00E2,U+00E3,U+00E4,U+00E5,U+0101,U+0103,U+0105  ##  à LATIN SMALL LETTER A WITH GRAVE
  U+00E1;U+0061;U+0061,U+00E0,U+00E2,U+00E3,U+00E4,U+00E5,U+0101,U+0103,U+0105  ##  á LATIN SMALL LETTER A WITH ACUTE
  U+00E2;U+0061;U+0061,U+00E0,U+00E1,U+00E3,U+00E4,U+00E5,U+0101,U+0103,U+0105  ##  â LATIN SMALL LETTER A WITH CIRCUMFLEX
  U+00E3;U+0061;U+0061,U+00E0,U+00E1,U+00E2,U+00E4,U+00E5,U+0101,U+0103,U+0105  ##  ã LATIN SMALL LETTER A WITH TILDE
  U+00E4;U+0061;U+0061,U+00E0,U+00E1,U+00E2,U+00E3,U+00E5,U+0101,U+0103,U+0105  ##  ä LATIN SMALL LETTER A WITH DIAERESIS
  U+00E5;U+0061;U+0061,U+00E0,U+00E1,U+00E2,U+00E3,U+00E4,U+0101,U+0103,U+0105  ##  å LATIN SMALL LETTER A WITH RING ABOVE
  U+00E6;U+0061 U+0065 ##  æ LATIN SMALL LETTER AE
  U+00E7;U+0063;U+0063,U+0107,U+0109,U+010B,U+010D  ##  ç LATIN SMALL LETTER C WITH CEDILLA
  U+00E8;U+0065;U+0065,U+00E9,U+00EA,U+00EB,U+0113,U+0115,U+0117,U+0119,U+011B  ##  è LATIN SMALL LETTER E WITH GRAVE
  U+00E9;U+0065;U+0065,U+00E8,U+00EA,U+00EB,U+0113,U+0115,U+0117,U+0119,U+011B  ##  é LATIN SMALL LETTER E WITH ACUTE
  U+00EA;U+0065;U+0065,U+00E8,U+00E9,U+00EB,U+0113,U+0115,U+0117,U+0119,U+011B  ##  ê LATIN SMALL LETTER E WITH CIRCUMFLEX
  U+00EB;U+0065;U+0065,U+00E8,U+00E9,U+00EA,U+0113,U+0115,U+0117,U+0119,U+011B  ##  ë LATIN SMALL LETTER E WITH DIAERESIS
  U+00EC;U+0069;U+0069,U+00ED,U+00EE,U+00EF,U+0129,U+012B,U+012D,U+012F,U+0131  ##  ì LATIN SMALL LETTER I WITH GRAVE
  U+00ED;U+0069;U+0069,U+00EC,U+00EE,U+00EF,U+0129,U+012B,U+012D,U+012F,U+0131  ##  í LATIN SMALL LETTER I WITH ACUTE
  U+00EE;U+0069;U+0069,U+00EC,U+00ED,U+00EF,U+0129,U+012B,U+012D,U+012F,U+0131  ##  î LATIN SMALL LETTER I WITH CIRCUMFLEX
  U+00EF;U+0069;U+0069,U+00EC,U+00ED,U+00EE,U+0129,U+012B,U+012D,U+012F,U+0131  ##  ï LATIN SMALL LETTER I WITH DIAERESIS
  U+00F0;U+0064;U+0064,U+00FE,U+010F,U+0111  ##  ð LATIN SMALL LETTER ETH
  U+00F1;U+006E;U+006E,U+0144,U+0146,U+0148,U+014B  ##  ñ LATIN SMALL LETTER N WITH TILDE
  U+00F2;U+006F;U+006F,U+00F3,U+00F4,U+00F5,U+00F6,U+00F8,U+014D,U+014F,U+0151  ##  ò LATIN SMALL LETTER O WITH GRAVE
  U+00F3;U+006F;U+006F,U+00F2,U+00F4,U+00F5,U+00F6,U+00F8,U+014D,U+014F,U+0151  ##  ó LATIN SMALL LETTER O WITH ACUTE
  U+00F4;U+006F;U+006F,U+00F2,U+00F3,U+00F5,U+00F6,U+00F8,U+014D,U+014F,U+0151  ##  ô LATIN SMALL LETTER O WITH CIRCUMFLEX
  U+00F5;U+006F;U+006F,U+00F2,U+00F3,U+00F4,U+00F6,U+00F8,U+014D,U+014F,U+0151  ##  õ LATIN SMALL LETTER O WITH TILDE
  U+00F6;U+006F;U+006F,U+00F2,U+00F3,U+00F4,U+00F5,U+00F8,U+014D,U+014F,U+0151  ##  ö LATIN SMALL LETTER O WITH DIAERESIS
  U+00F8;U+006F;U+006F,U+00F2,U+00F3,U+00F4,U+00F5,U+00F6,U+014D,U+014F,U+0151  ##  ø LATIN SMALL LETTER O WITH STROKE
  U+00F9;U+0075;U+0075,U+00FA,U+00FB,U+00FC,U+0169,U+016B,U+016D,U+016F,U+0171,U+0173  ##  ù LATIN SMALL LETTER U WITH GRAVE
  U+00FA;U+0075;U+0075,U+00F9,U+00FB,U+00FC,U+0169,U+016B,U+016D,U+016F,U+0171,U+0173  ##  ú LATIN SMALL LETTER U WITH ACUTE
  U+00FB;U+0075;U+0075,U+00F9,U+00FA,U+00FC,U+0169,U+016B,U+016D,U+016F,U+0171,U+0173  ##  û LATIN SMALL LETTER U WITH
CIRCUMFLEX
  U+00FC;U+0075;U+0075,U+00F9,U+00FA,U+00FB,U+0169,U+016B,U+016D,U+016F,U+0171,U+0173  ##  ü LATIN SMALL LETTER U WITH
DIAERESIS
  U+00FD;U+0079;U+0079,U+00FF,U+0177  ##  ý LATIN SMALL LETTER Y WITH ACUTE
  U+00FE;U+0064;U+0064,U+00F0,U+010F,U+0111  ##  þ LATIN SMALL LETTER THORN
  U+00FF;U+0079;U+0079,U+00FD,U+0177  ##  ÿ LATIN SMALL LETTER Y WITH DIAERESIS
######
###### Latin Extended-A
  U+0101;U+0061;U+0061,U+00E0,U+00E1,U+00E2,U+00E3,U+00E4,U+00E5,U+0103,U+0105  ##  ā LATIN SMALL LETTER A WITH MACRON
  U+0103;U+0061;U+0061,U+00E0,U+00E1,U+00E2,U+00E3,U+00E4,U+00E5,U+0101,U+0105  ##  ă LATIN SMALL LETTER A WITH BREVE
  U+0105;U+0061;U+0061,U+00E0,U+00E1,U+00E2,U+00E3,U+00E4,U+00E5,U+0101,U+0103  ##  ą LATIN SMALL LETTER A WITH OGONEK
  U+0107;U+0063;U+0063,U+00E7,U+0109,U+010B,U+010D  ##  ć LATIN SMALL LETTER C WITH ACUTE
  U+0109;U+0063;U+0063,U+00E7,U+0107,U+010B,U+010D  ##  ĉ LATIN SMALL LETTER C WITH CIRCUMFLEX
  U+010B;U+0063;U+0063,U+00E7,U+0107,U+0109,U+010D  ##  ċ LATIN SMALL LETTER C WITH DOT ABOVE
  U+010D;U+0063;U+0063,U+00E7,U+0107,U+0109,U+010B  ##  č LATIN SMALL LETTER C WITH CARON
  U+010F;U+0064;U+0064,U+00F0,U+00FE,U+0111  ##  ď LATIN SMALL LETTER D WITH CARON
  U+0111;U+0064;U+0064,U+00F0,U+00FE,U+010F  ##  đ LATIN SMALL LETTER D WITH STROKE
  U+0113;U+0065;U+0065,U+00E8,U+00E9,U+00EA,U+00EB,U+0115,U+0117,U+0119,U+011B  ##  ē LATIN SMALL LETTER E WITH MACRON
  U+0115;U+0065;U+0065,U+00E8,U+00E9,U+00EA,U+00EB,U+0113,U+0117,U+0119,U+011B  ##  ĕ LATIN SMALL LETTER E WITH BREVE
  U+0117;U+0065;U+0065,U+00E8,U+00E9,U+00EA,U+00EB,U+0113,U+0115,U+0119,U+011B  ##  ė LATIN SMALL LETTER E WITH DOT ABOVE
  U+0119;U+0065;U+0065,U+00E8,U+00E9,U+00EA,U+00EB,U+0113,U+0115,U+0117,U+011B  ##  ę LATIN SMALL LETTER E WITH OGONEK
  U+011B;U+0065;U+0065,U+00E8,U+00E9,U+00EA,U+00EB,U+0113,U+0115,U+0117,U+0119  ##  ě LATIN SMALL LETTER E WITH CARON
  U+011D;U+0067;U+0067,U+011F,U+0121,U+0123  ##  ĝ LATIN SMALL LETTER G WITH CIRCUMFLEX
  U+011F;U+0067;U+0067,U+011D,U+0121,U+0123  ##  ğ LATIN SMALL LETTER G WITH BREVE
  U+0121;U+0067;U+0067,U+011D,U+011F,U+0123  ##  ġ LATIN SMALL LETTER G WITH DOT ABOVE
  U+0123;U+0067;U+0067,U+011D,U+011F,U+0121  ##  ģ LATIN SMALL LETTER G WITH CEDILLA
  U+0125;U+0068;U+0068,U+0127  ##  ĥ LATIN SMALL LETTER H WITH CIRCUMFLEX
  U+0127;U+0068;U+0068,U+0125  ##  ħ LATIN SMALL LETTER H WITH STROKE
  U+0129;U+0069;U+0069,U+00EC,U+00ED,U+00EE,U+00EF,U+012B,U+012D,U+012F,U+0131  ##  ĩ LATIN SMALL LETTER I WITH TILDE
  U+012B;U+0069;U+0069,U+00EC,U+00ED,U+00EE,U+00EF,U+0129,U+012D,U+012F,U+0131  ##  ī LATIN SMALL LETTER I WITH MACRON
  U+012D;U+0069;U+0069,U+00EC,U+00ED,U+00EE,U+00EF,U+0129,U+012B,U+012F,U+0131  ##  ĭ LATIN SMALL LETTER I WITH BREVE
  U+012F;U+0069;U+0069,U+00EC,U+00ED,U+00EE,U+00EF,U+0129,U+012B,U+012D,U+0131  ##  į LATIN SMALL LETTER I WITH OGONEK
  U+0131;U+0069;U+0069,U+00EC,U+00ED,U+00EE,U+00EF,U+0129,U+012B,U+012D,U+012F  ##  ı LATIN SMALL LETTER DOTLESS I
  U+0135;U+006A;U+006A  ##  ĵ LATIN SMALL LETTER J WITH CIRCUMFLEX
  U+0137;U+006B;U+006B  ##  ķ LATIN SMALL LETTER K WITH CEDILLA
  U+013A;U+006C;U+006C,U+013C,U+013E,U+0142  ##  ĺ LATIN SMALL LETTER L WITH ACUTE
  U+013C;U+006C;U+006C,U+013A,U+013E,U+0142  ##  ļ LATIN SMALL LETTER L WITH CEDILLA
  U+013E;U+006C;U+006C,U+013A,U+013C,U+0142  ##  ľ LATIN SMALL LETTER L WITH CARON
  U+0142;U+006C;U+006C,U+013A,U+013C,U+013E  ##  ł LATIN SMALL LETTER L WITH STROKE
  U+0144;U+006E;U+006E,U+00F1,U+0146,U+0148,U+014B  ##  ń LATIN SMALL LETTER N WITH ACUTE
  U+0146;U+006E;U+006E,U+00F1,U+0144,U+0148,U+014B  ##  ņ LATIN SMALL LETTER N WITH CEDILLA
  U+0148;U+006E;U+006E,U+00F1,U+0144,U+0146,U+014B  ##  ň LATIN SMALL LETTER N WITH CARON
  U+014B;U+006E;U+006E,U+00F1,U+0144,U+0146,U+0148  ##  ŋ LATIN SMALL LETTER ENG
  U+014D;U+006F;U+006F,U+00F2,U+00F3,U+00F4,U+00F5,U+00F6,U+00F8,U+014F,U+0151  ##  ō LATIN SMALL LETTER O WITH MACRON
  U+014F;U+006F;U+006F,U+00F2,U+00F3,U+00F4,U+00F5,U+00F6,U+00F8,U+014D,U+0151  ##  ŏ LATIN SMALL LETTER O WITH BREVE
  U+0151;U+006F;U+006F,U+00F2,U+00F3,U+00F4,U+00F5,U+00F6,U+00F8,U+014D,U+014F  ##  ő LATIN SMALL LETTER O WITH DOUBLE ACUTE
  U+0153;U+006F U+0065  ##  œ LATIN SMALL LIGATURE OE
  U+0155;U+0072;U+0072,U+0157,U+0159  ##  ŕ LATIN SMALL LETTER R WITH ACUTE
  U+0157;U+0072;U+0072,U+0155,U+0159  ##  ŗ LATIN SMALL LETTER R WITH CEDILLA
  U+0159;U+0072;U+0072,U+0155,U+0157  ##  ř LATIN SMALL LETTER R WITH CARON
  U+015B;U+0073;U+0073,U+015D,U+0161,U+0219  ##  ś LATIN SMALL LETTER S WITH ACUTE
  U+015D;U+0073;U+0073,U+015B,U+0161,U+0219  ##  ŝ LATIN SMALL LETTER S WITH CIRCUMFLEX
  U+0161;U+0073;U+0073,U+015B,U+015D,U+0219  ##  š LATIN SMALL LETTER S WITH CARON
  U+0165;U+0074;U+0074,U+0167,U+021B  ##  ť LATIN SMALL LETTER T WITH CARON
  U+0167;U+0074;U+0074,U+0165,U+021B  ##  ŧ LATIN SMALL LETTER T WITH STROKE
  U+0169;U+0075;U+0075,U+00F9,U+00FA,U+00FB,U+00FC,U+016B,U+016D,U+016F,U+0171,U+0173  ##  ũ LATIN SMALL LETTER U WITH TILDE
  U+016B;U+0075;U+0075,U+00F9,U+00FA,U+00FB,U+00FC,U+0169,U+016D,U+016F,U+0171,U+0173  ##  ū LATIN SMALL LETTER U WITH MACRON
  U+016D;U+0075;U+0075,U+00F9,U+00FA,U+00FB,U+00FC,U+0169,U+016B,U+016F,U+0171,U+0173  ##  ŭ LATIN SMALL LETTER U WITH BREVE
  U+016F;U+0075;U+0075,U+00F9,U+00FA,U+00FB,U+00FC,U+0169,U+016B,U+016D,U+0171,U+0173  ##  ů LATIN SMALL LETTER U WITH RING
ABOVE
  U+0171;U+0075;U+0075,U+00F9,U+00FA,U+00FB,U+00FC,U+0169,U+016B,U+016D,U+016F,U+0173  ##  ű LATIN SMALL LETTER U WITH DOUBLE
ACUTE
  U+0173;U+0075;U+0075,U+00F9,U+00FA,U+00FB,U+00FC,U+0169,U+016B,U+016D,U+016F,U+0171  ##  ų LATIN SMALL LETTER U WITH OGONEK
  U+0175;U+0077;U+0077  ##  ŵ LATIN SMALL LETTER W WITH CIRCUMFLEX
  U+0177;U+0079;U+0079,U+00FD,U+00FF  ##  ŷ LATIN SMALL LETTER Y WITH CIRCUMFLEX
  U+017A;U+007A;U+007A,U+017C,U+017E  ##  ź LATIN SMALL LETTER Z WITH ACUTE
  U+017C;U+007A;U+007A,U+017A,U+017E  ##  ż LATIN SMALL LETTER Z WITH DOT ABOVE
  U+017E;U+007A;U+007A,U+017A,U+017C  ##  ž LATIN SMALL LETTER Z WITH CARON
######
###### Latin Extended-B
  U+0219;U+0073;U+0073,U+015B,U+015D,U+0161  ##  ș LATIN SMALL LETTER S WITH COMMA BELOW
  U+021B;U+0074;U+0074,U+0165,U+0167  ##  ț LATIN SMALL LETTER T WITH COMMA BELOW
```

main:CommonEPPPoliciesText: 20231215-123350
main:CommonIDNExtensionText: 20211126-082810
core:SwissLPText: 20150819-131715
main:CommonQ44IDNTablesText: 20211125-111542
main:CommonContactEligibilityExtension11Text: 20240115-101516
main:CommonQ44IDNTableLatinScript: 20131126-104856
main:CommonContactEligibilityExtensionText: 20231201-174618
main:CommonRegistryRegistrarDocumentation: 20231215-124632
main:CommonMarkDataExtensionText: 20231215-124219
core:LaunchPhaseSpecificationsText: 20231113-151938
core:SwissDef: 20230830-112135
main:CommonDefinitions: 20231201-121455
main:CommonDefaults: 20230928-112358
main:CommonRegistrar: 20150319-174249