RFC 9022 : Domain Name Registration Data (DNRD) Objects Mapping

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 30 mai 2021

Date de publication du RFC : Mai 2021

https://www.bortzmeyer.org/9022.html

Le RFC 8909 [1] normalisait un format générique pour les séquestres d'objets enregistrés dans un registre. Ce nouveau RFC 9022 précise ce format pour le cas spécifique des registres de noms de domaine (dont les objets enregistrés sont les noms de domaine, les contacts, les serveurs de noms, etc).

Rappelons que le but d'un séquestre est de permettre à un tiers de reprendre les opérations d'un registre (par exemple un registre de noms de domaine) en cas de défaillance complète de celui-ci. Pas juste une panne matérielle qui fait perdre des données, non, une défaillance grave, par exemple une faillite, qui fait que plus rien ne marche. Le registre doit donc envoyer régulièrement des données à l'opérateur de séquestre qui, au cas où la catastrophe survient, enverra ces données au nouveau registre, qui sera en mesure (en théorie…) de charger les données dans une nouvelle base et de reprendre les opérations. Sous quel format sont envoyées ces données ? Car il faut évidemment un format ouvert et documenté, pour que le nouveau registre ait pu développer un programme d'importation des données. C'est le but du RFC 8909 et de notre nouveau RFC 9022. Ils spécifient un format à base de XML (avec possibilité de CSV).

Par rapport au format générique du RFC 8909, notre RFC ajoute les objets spécifiques de l'industrie des noms de domaine (il est recommandé de réviser la terminologie du RFC 8499) :

— Les noms de domaine, tels que manipulés en EPP selon le RFC 5731.
— Les serveurs de noms (pour les registres où ils sont enregistrés comme objets séparés, ce qui n'est pas obligatoire), tels que gérés via EPP selon le RFC 5732.
— Les contacts (le titulaire du nom de domaine, le contact technique avec qui on verra les problèmes DNS, etc), selon le modèle utilisé en EPP par le RFC 5733.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc8909.txt

— Les BE (Bureaux d'Enregistrement). Le cas est un peu différent car ils ne sont typiquement pas gérés via EPP puisque les sessions EPP sont justement établies par un BE, qui a donc dû être créé par un autre moyen.
— Les noms spéciaux, qui sont des noms de domaine qui ne sont pas des noms de domaines habituels mais, par exemple, des gros mots dont l'enregistrement est interdit. En anglais, le sigle officiel est NNDN, qui veut récursivement dire "*NNDN's Not a Domain Name*".
— Les tables liées à IDN, pour les registres qui maintiennent des informations restreignant l'usage de ces noms de domaine en Unicode (cf. RFC 6927).
— Certains paramètres de configuration des serveurs EPP du registre.
— Des compteurs du nombre d'objets enregistrés.
— Des objets qui sont importants pour le registre mais pas assez répandus pour avoir fait l'objet d'une normalisation.

Le format concret du séquestre se décline en deux « modèles », un en XML et un en CSV (RFC 4180), mais j'ai l'impression que le XML est nettement plus utilisé. Dans tous les cas :

— Les dates sont au format du RFC 3339.
— Les noms des pays sont les identificateurs à deux lettres de ISO 3166.
— Les numéros de téléphone sont au format E.164.
— Les adresses IP doivent suivre le format du RFC 5952. Pour IPv4, le RFC est vague, car il n'y a pas de format standard (vous ne verrez pas une seule adresse IP dans le RFC 791, que cite notre RFC…).

Pour CSV, chaque fichier CSV représente une table (les noms de domaine, les contacts, les serveurs de noms…) et chaque ligne du fichier un objet. Le modèle de données est décrit plus précisément en section 4.6. Voici un exemple d'une ligne extraite du fichier des noms de domaine, décrivant le domaine `domain1.example`, créé le 3 avril 2009, et dont le titulaire a pour identificateur ("*handle*") `registrantid` :

```
domain1.example,Ddomain2-TEST,,,registrantid,registrarX,registrarX,clientY,2009-04-03T22:00:00.0Z,registrarX
```

Les contacts seraient mis dans un autre fichier CSV, avec un fichier de jointure pour faire le lien entre domaines et contacts.

Et pour XML ? Il s'inspire beaucoup des éléments XML échangés avec EPP, par exemple pour la liste des états possibles pour un domaine. Voici un exemple (RDE = "*Registry Data Escrow*", et l'espace de noms correspondant `rdeDom` est `urn:ietf:params:xml:ns:rdeDomain-1.0`) :

```
<rdeDom:domain>
      <rdeDom:name>jdoe.example</rdeDom:name>
      <rdeDom:roid>DOM03-EXAMPLE</rdeDom:roid>
      <rdeDom:status s="ok"/>
      <rdeDom:registrant>IZT01</rdeDom:registrant>
      <rdeDom:contact type="tech">IZT01</rdeDom:contact>
      <rdeDom:contact type="billing">IZT01</rdeDom:contact>
      <rdeDom:contact type="admin">IZT01</rdeDom:contact>
      <rdeDom:ns>
        <domain:hostObj>ns.jdoe.example</domain:hostObj>
      </rdeDom:ns>
      <rdeDom:clID>RAR03</rdeDom:clID>
      <rdeDom:crRr>RAR03</rdeDom:crRr>
      <rdeDom:crDate>2019-12-26T14:18:40.65647Z</rdeDom:crDate>
      <rdeDom:exDate>2020-12-26T14:18:40.509742Z</rdeDom:exDate>
</rdeDom:domain>
```

On voit que cela ressemble en effet beaucoup à ce qui avait été envoyé en EPP pour créer le domaine (cf. RFC 5731). Si vous voulez un exemple complet et réaliste, regardez les sections 14 et 15 du RFC.

Et voici un exemple de contact (RFC 5733) :

```
<rdeContact:contact>
      <rdeContact:id>IZT01</rdeContact:id>
      <rdeContact:status s="ok"/>
      <rdeContact:postalInfo type="loc">
        <contact:name>John Doe</contact:name>
        <contact:addr>
          <contact:street>12 Rue de la Paix</contact:street>
          <contact:city>Paris</contact:city>
          <contact:pc>75002</contact:pc>
          <contact:cc>FR</contact:cc>
        </contact:addr>
      </rdeContact:postalInfo>
      <rdeContact:voice>+33.0353011234</rdeContact:voice>
      <rdeContact:email>john.doe@foobar.example</rdeContact:email>
      <rdeContact:clID>RAR03</rdeContact:clID>
      <rdeContact:crRr>RAR03</rdeContact:crRr>
      <rdeContact:crDate>2019-12-26T13:47:05.580392Z</rdeContact:crDate>
      <rdeContact:disclose flag="0">
        <contact:name type="loc"/>
        <contact:addr type="loc"/>
        <contact:voice/>
        <contact:fax/>
        <contact:email/>
      </rdeContact:disclose>
</rdeContact:contact>
```

On notera l'élement `<disclose>` qui indique qu'on ne doit pas diffuser le nom, l'adresse ou d'autres éléments sur le contact (normal, il s'agit d'une personne physique, et la loi Informatique & Libertés s'applique, cf. section 14 du RFC). La jointure avec les domaines dont il est contact (comme le `jdoe.example` plus haut), se fait sur l'identificateur (élément `<id>`, dit aussi "*handle*"). L'information sur l'adresse a le type `loc`, ce qui veut dire qu'elle peut utiliser tout le jeu de caractères Unicode. Avec le type `int`, elle serait restreinte à l'ASCII (une très ancienne erreur fait que EPP appelle `loc` - "*local*", ce qui est internationalisé et `int` - "*international*" ce qui est restreint aux lettres utilisées en anglais).

Et enfin, un objet représentant un serveur de noms (RFC 5732) :

```
<rdeHost:host>
      <rdeHost:name>ns1.foobar.example</rdeHost:name>
      <rdeHost:status s="ok"/>
      <rdeHost:addr ip="v6">2001:db8:cafe:fada::53</rdeHost:addr>
      <rdeHost:clID>RAR02</rdeHost:clID>
      <rdeHost:crRr>RAR02</rdeHost:crRr>
      <rdeHost:crDate>2020-05-13T12:37:41.788684Z</rdeHost:crDate>
</rdeHost:host>
```

Ce format de séquestre permet aussi de représenter des objets qui n'ont pas d'équivalent en EPP, comme les bureaux d'enregistrement, qui ne peuvent pas être créés en EPP puisque la session EPP est liée au client du registre, donc au bureau d'enregistrement. Un exemple de BE (Bureau d'Enregistrement) :

```
<rdeRegistrar:registrar>
      <rdeRegistrar:id>RAR21</rdeRegistrar:id>
      <rdeRegistrar:name>Name Business</rdeRegistrar:name>
      <rdeRegistrar:status>ok</rdeRegistrar:status>
      <rdeRegistrar:postalInfo type="loc">
        <rdeRegistrar:addr>
          <rdeRegistrar:street>1 rue du Test</rdeRegistrar:street>
          <rdeRegistrar:city>Champignac</rdeRegistrar:city>
          <rdeRegistrar:cc>FR</rdeRegistrar:cc>
        </rdeRegistrar:addr>
      </rdeRegistrar:postalInfo>
      <rdeRegistrar:voice>+33.0639981234</rdeRegistrar:voice>
      <rdeRegistrar:fax>+33.0199001234</rdeRegistrar:fax>
      <rdeRegistrar:email>master-of-domains@namebusiness.example</rdeRegistrar:email>
</rdeRegistrar:registrar>
```

On peut aussi mettre dans le séquestre des références vers ses tables IDN (que l'ICANN exige mais qui n'ont aucun intérêt). Plus intéressant, la possibilité de stocker dans le séquestre les listes de termes traités spécialement, par exemple interdits ou bien soumis à un examen manuel `<https://www.afnic.fr/observatoire-ressources/documents/charte-nommage/>`. Cela se nomme NNDN pour « *"NNDN's not domain name"* », oui, c'est récursif. Voici un exemple :

```
<rdeNNDN:NNDN>
     <rdeNNDN:uName>gros-mot.example</rdeNNDN:uName>
     <rdeNNDN:nameState>blocked</rdeNNDN:nameState>
     <rdeNNDN:crDate>2005-04-23T11:49:00.0Z</rdeNNDN:crDate>
   </rdeNNDN:NNDN>
```

Tous les registres n'ont pas les mêmes règles et le RFC décrit également les mécanismes qui permettent de spécifier dans le séquestre les contraintes d'intégrité spécifiques d'un registre. L'opérateur de séquestre, qui reçoit le fichier XML ou les fichiers CSV, est censé vérifier tout cela (autrement, il ne joue pas son rôle, s'il se contente de stocker aveuglément un fichier). La section 8 de notre RFC décrit plus en profondeur les vérifications recommandées, comme de vérifier que les contacts indiqués pour chaque domaine sont bien présents. Pour vérifier un séquestre, il faut importer beaucoup de schémas. Voici, la liste, sous forme d'une commande shell :

```
for schema in contact-1.0.xsd  host-1.0.xsd rdeDomain-1.0.xsd     rdeIDN-1.0.xsd     rgp-1.0.xsd domain-1.0.
rde-1.0.xsd rdeEppParams-1.0.xsd  rdeNNDN-1.0.xsd     secDNS-1.1.xsd epp-1.0.xsd  rdeContact-1.0.xsd
rdeHeader-1.0.xsd     rdePolicy-1.0.xsd   eppcom-1.0.xsd  rdeDnrdCommon-1.0.xsd rdeHost-1.0.xsd       rdeReg
   wget https://www.iana.org/assignments/xml-registry/schema/${schema}
done
```

Ensuite, on importe (fichier (en ligne sur `https://www.bortzmeyer.org/files/escrow-wrapper.xsd`)) et on utilise xmllint sur l'exemple de séquestre de la section 14 du RFC (fichier (en ligne sur `https://www.bortzmeyer.org/files/escrow-example.xml`)) :

```
% xmllint --noout --schema wrapper.xsd escrow-example.xml
escrow-example.xml validates
```

Ouf, tout va bien, le registre nous a envoyé un séquestre correct.

Enfin, la syntaxe formelle de ce format figure dans la section 9 du RFC, dans le langage XML Schema.

Ce format est mis en œuvre par tous les TLD qui sont liés par un contrat avec l'ICANN. 1 200 TLD <https://www.icann.org/en/registry-agreements> envoient ainsi un séquestre une fois par semaine à l'ICANN.

Le concept de séquestre pose de sérieux problèmes de sécurité car le ou les fichiers transmis sont typiquement à la fois confidentiels, et cruciaux pour assurer la continuité de service. Lors du transfert du fichier, le registre et l'opérateur de séquestre doivent donc vérifier tous les deux l'authenticité du partenaire, et la confidentialité de la transmission. D'autant plus qu'une bonne partie du fichier est composée de données personnelles.