Appendix B - Registrar Testing
Testing CORE GatewayNG Registrar System

1 Testing the Control Panel (Web Interface) - G4, G5

1.1 Contact

1.1.1 Summary
1.1.2 Detailed Test Descriptions
1.1.2.1 Create:
1.1.2.2 Update
1.1.2.3 Search

1.2 Host

1.2.1 Summary
1.2.2 Detailed Test Descriptions
1.2.2.1 Create
1.2.2.2 Search
1.2.2.3 Update

1.3.1 Summary
1.3.2 Detailed Test Descriptions
1.3.2.1 Create
1.3.2.2 Edit (add variants)
1.3.2.3 Edit (add hosts)
1.3.2.4 Search
1.3.2.5 Check (existing variant)
1.3.2.6 Check (not registered variant)
1.3.2.7 Check (unregistered label)
1.3.2.8 Transfer
1.3.2.9 Search transfer

2 Testing the Payload Interface - G5

2.1 Contact

2.1.1 Summary
2.1.2 Detailed Test Descriptions
2.1.2.1 Create
2.1.2.2 Info
2.1.2.3 Update
2.1.2.4 Update (adjusted) 19

2.2 Host 20
  2.2.1 Summary 20
  2.2.2 Detailed Test Descriptions 21
    2.2.2.1 Create (U-label) 21
    2.2.2.2 Create (A-label) 22
    2.2.2.3 Info 23
    2.2.2.4 Update 23

2.2.2.1 Summary 21
  2.2.2.2 Detailed Test Descriptions 22
    2.2.2.2.1 Create (U-label) 22
    2.2.2.2.2 Create (A-label) 23
    2.2.2.2.3 Info 24
    2.2.2.2.4 Update (setting variants) 25
    2.2.2.2.5 Check (variant) 26
    2.2.2.2.6 Check (non-variant) 27
    2.2.2.2.7 Transfer 28

2.3 Domain 24
  2.3.1 Summary 24
  2.3.2 Detailed Test Descriptions 25
    2.3.2.1 Create (U-label) 25
    2.3.2.2 Create (A-label) 26
    2.3.2.3 Info 27
    2.3.2.4 Update (setting variants) 28
    2.3.2.5 Check (variant) 29
    2.3.2.6 Check (non-variant) 30
    2.3.2.7 Transfer 31

3 Testing the Authoritative DNS Server - G1 32
  3.1 DNS Query 32
    3.1.1 Detailed Test Description 32

4 Testing the Port 43 Whois Interface - G2 34
  4.1 Contact 34
    4.1.1 Detailed Test Description 34
  4.2 Host 35
    4.2.1 Detailed Test Description 35
  4.3 Domain 36
    4.3.1 Detailed Test Description 36

5 Testing the RDAP Interface - G3 37
  5.1 Contact 37
    5.1.1 Detailed Test Description 37
  5.2 Host 39
    5.2.1 Detailed Test Description 39
Testing CORE GatewayNG Registrar System

The test cases described in detail in this Appendix do not cover all of the gates identified in the main document. It merely serves as an approach to test the registrar’s main features that are easily accessible without internal knowledge. As such we are providing sample test cases for the UA-readiness gates G1, G2, G3, G4, G5, and G6 thereby implicitly also testing the internal gates G7, G8, G9, and G11.

In order to make it easier for the reader to find their way in the test cases, colour codes have been used. For each test case the expected outcome is stated followed by either a green success or red fail keyword. Furthermore, the respective output of the test case has been marked with a green or red background to easily see where the result is expected to be found.

This appendix is part of the following three-part report:

1. Universal Acceptance (UA) Roadmap for Domain Name Registry and Registrar Systems.
2. Appendix A - Registry Testing
3. Appendix B - Registrar Testing
1 Testing the Control Panel (Web Interface) - G4, G5

The test cases within this section reflect the UA-readiness gates G4 and G5 as visualized in Figure 2: Domain Registrar High-Level System Architecture in the main document. Testing a registrar’s web interface will be different for each registrar system as there is no common standard and registrars are free to offer any kind of features and functionality. Nevertheless, it is likely that standard functionality (create, read, update, delete) will be available. Together with the optional searching/filtering of objects, these will be the test cases we are looking at.

1.1 Contact

1.1.1 Summary

<table>
<thead>
<tr>
<th>Operation</th>
<th>Outcome</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>fail</td>
<td>δοκιμή@テスト.μημεκάθε incorrectly flagged as invalid</td>
</tr>
<tr>
<td>Create (adjusted)</td>
<td>fail</td>
<td>Used 测试@测试.测试 as email address</td>
</tr>
<tr>
<td>Create (adjusted)</td>
<td>success</td>
<td>Used <a href="mailto:example@xn--0zwm56d.xn--0zwm56d">example@xn--0zwm56d.xn--0zwm56d</a> as email address</td>
</tr>
<tr>
<td>Update</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Search</td>
<td>partial success</td>
<td>Domain name in email address is not normalized to U-label notation; searching by using the A-label version (测试@xn--0zwm56d.xn--0zwm56d) does not find the contact</td>
</tr>
<tr>
<td>View</td>
<td>success</td>
<td>Domain name in email address is not normalized to U-label notation, it is displayed the way it was created</td>
</tr>
</tbody>
</table>

For the Contact test cases only the email address was of concern. Any other data, which may contain the full range of Unicode characters (e.g., internationalized address data), is ignored in the context of Universal Acceptance.

It should be tested that an EAI is accepted when storing and updating contacts and that the email address is correctly displayed when viewing the contact. In cases where searching in the contact space is supported, the contact must be found when specifying the email address. Care should be taken that the domain name of the email address is supported in A-label as well as U-label format and both formats are considered as equal when searching by email address. While this list of tests is intended to be a reasonably complete list, there is no guarantee for completeness and there may be other checks depending on the actually deployed software.
Testing the GatewayNG registrar system resulted in one problem:

1. No EAI was accepted, neither δοκιμή@テス ト.みんな@dora nor 测试@测试.测试 nor <example@grün.de> nor grün@example.com was accepted. The web interface uses vue.js [vue] and the vuelidate [vuelidate] library 0.7.7. The library uses a regular expression for validation, which does not support any non-ASCII character. Starting with release 1.59 the GatewayNG software will switch to a simple regular expression, leaving the more complex validation to the backend Java code.

1.1.2 Detailed Test Descriptions

1.1.2.1 Create:

Enter δοκιμή@テス ト.みんな@dora as email address.

Expected outcome:
- Storage is possible: fail, email validation in Frontend fails
- Display of contact shows the same values: unable to check

Alter test to use different email address: 测试@测试.测试

Expected outcome:
- Storage is possible: fail, email validation in Frontend fails
- Display of contact shows the same values: unable to check

Alter test to use a ASCII-only email address: example@xn--0zwm56d.xn--0zwm56d

Expected outcome:
- Storage is possible: success
- Display of contact shows the same values: success

1.1.2.2 Update

Same behaviour as create.
1.1.2.3 Search
Search the contacts by
  - Email "example@xn--0zwm56d.xn--0zwm56d"
Expected outcome:
  - Find the contact: success

Search contact by
  - Email "example@测试.测试"

Expected outcome:
  - Find the contact: fail
1.2 Host

1.2.1 Summary

<table>
<thead>
<tr>
<th>Operation</th>
<th>Outcome</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>success</td>
<td>Both A-label and U-label are supported</td>
</tr>
<tr>
<td>Update</td>
<td>n/a</td>
<td>No renaming of hosts is allowed.</td>
</tr>
<tr>
<td>Search</td>
<td>success</td>
<td>Searching by A-label also finds hosts created using U-labels and vice versa</td>
</tr>
<tr>
<td>View</td>
<td>success</td>
<td>Both A-label and U-label domain names are displayed at the same time</td>
</tr>
</tbody>
</table>

It should be tested that hosts are accepted in either A-label or U-label notation. In case searching in the host space is supported, the host must be found when specifying the host’s domain name. Care should be taken that the domain name of the host is supported in A-label as well as U-label format and both formats are considered as equal when searching by domain name. It is suggested to store the host’s domain name in a normalised form.

Testing the GatewayNG registrar system showed full support of A-label and U-label. For input, either of the two notations is accepted; for output always both versions are displayed next to each other. When searching for hosts, there is a single input field to enter the A-label or the U-label domain name, for both notations also a prefix search is supported.

1.2.2 Detailed Test Descriptions

1.2.2.1 Create

Enter “پارکینگی” as host name.

Expected outcome:
- Storage is possible: success
- Display of host shows the same domain name: success
- Display of A-labels of the host shows as “xn--kgbechtv.xn--hgbk6aj7f53bba.xn--11b5bs3a9aj6g”: success
1.2.2.2 Search

Search hosts by U-label "اختبار.آزمایشی" and by A-label "xn--kgbechtv.xn--hgbk6aj7f53bba.xn--11b5bs3a9aj6g"

Expected outcome:
- Find the corresponding host with U-label search: success
- Find the corresponding host with A-label search: success

1.2.2.3 Update

System does not allow renaming of hosts in general.
1.3 Domain

1.3.1 Summary

<table>
<thead>
<tr>
<th>Operation</th>
<th>Outcome</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>success</td>
<td>Both A-label and U-label are supported</td>
</tr>
<tr>
<td>Update</td>
<td>success</td>
<td>Both A-label and U-label are supported</td>
</tr>
<tr>
<td>Update (add variants)</td>
<td>success</td>
<td>Both A-label and U-label are supported</td>
</tr>
<tr>
<td>Update (add hosts)</td>
<td>success</td>
<td>Hosts in A-label and U-label format are supported</td>
</tr>
<tr>
<td>Search</td>
<td>success</td>
<td>Searching by A-label also finds domains created using U-labels and vice versa</td>
</tr>
<tr>
<td>View</td>
<td>success</td>
<td>Both A-label and U-label domain names are displayed at the same time</td>
</tr>
<tr>
<td>Check</td>
<td>success</td>
<td>Checking by A-label and U-label give the same result</td>
</tr>
<tr>
<td>Transfer</td>
<td>success</td>
<td>Both A-label and U-label domain names are supported</td>
</tr>
</tbody>
</table>

It should be tested that domains are accepted in either A-label or U-label notation. Where searching in the domain space is supported, the domain must be found when specifying the domain name. Care should be taken that the domain name is supported in A-label as well as U-label format and both formats are considered as equal when searching by domain name.

Testing the GatewayNG registrar system showed full support of A-label and U-label. For input, either of the two notations is accepted; for output always both versions are displayed next to each other. When searching for domains, there is a single input field to enter either the A-label or the U-label domain name, for both notation also a prefix search is supported. Variants are supported based on the configured IDN table. A variant label may be added either in A-label or U-label notation as long as it constitutes an allocatable variant according to the IDN table. When checking domains, all existing domains as well as all variants (independent of their allocation status) are considered as blocked.
1.3.2 Detailed Test Descriptions

1.3.2.1 Create

Enter "sëst.テスト" as domain name (choosing French language tag “fr”).

Expected outcome:
- Storage is possible: success
- Display of domain shows the same domain name: success
- Display of A-label domain name shows as "xn--sst-jma.xn--zckzah": success

1.3.2.2 Edit (add variants)

Edit "sëst.テスト", add variants "sest.テスト" and "sést.テスト"

Expected outcome:
- Storage is possible: success
- Display of domain variant labels show the same values: success
- Display of A-labels of domain variant labels show as "sest.xn--zckzah” and "xn--sst-bma.xn--zckzah”, resp.: success

1.3.2.3 Edit (add hosts)

Enter "ns1.mylabel.பரிட்சை", "ns1.测试.பரிட்சை", and “ns2.mylabel.xn--hlcj6aya9esc7a” as host labels.

Expected outcome:
- Storage is possible: success
- Host names are normalised and their U-label version is correctly displayed: success

1.3.2.4 Search

Search domains by U-label “sëst.テスト” and by A-label “xn--sst-jma.xn--zckzah”

Expected outcome:
- Find the corresponding domain with U-label search: success
- Find the corresponding domain with A-label search: success
1.3.2.5 Check (existing variant)
Enter "tést.テスト" as domain name.

Expected outcome:
- Check is executed: success
- Check result is “not available”: success

1.3.2.6 Check (not registered variant)
Enter "tést.テスト" as domain name.

Expected outcome:
- Check is executed: success
- Check result is “not available”, blocked by variant: success

1.3.2.7 Check (unregistered label)
Enter "tost.テスト" as domain name.

Expected outcome:
- Check is executed: success
- Check result is “available”: success

1.3.2.8 Transfer
Transfer the domain “tést.テスト” to another registrar.

Expected outcome:
- Domain transfer is initiated: success

1.3.2.9 Search transfer
Search transfer by U-label “tést.テスト” and A-label “xn--tst-jma.xn--zckzah”

Expected outcome:
- Find the corresponding pending transfer with U-label search: success
- Find the corresponding pending transfer with A-label search: success
2 Testing the Payload Interface - G5

The test cases within this section reflect the UA-readiness gate G5 as visualized in Figure 2: Domain Registrar High-Level System Architecture in the main document. Whereas registries usually use a common standard for their automated interface, namely EPP (see also Appendix X: Registry Testing) there is no such standard for registrar systems. Some also employ EPP while others use SOAP, REST or their own proprietary API. The GatewayNG offers a simple API called Common Provisioning Protocol, historically often simply called Payload. The payload definition [GWPayload] specifies the syntax of requests accepted by the CORE GatewayNG and the corresponding responses.

The format consists of key/value pairs, where each pair is given on a separate input/output line of the request/response; key and value are separated by a colon. While this protocol is specific to CORE’s registrar system, the following examples can still be used to create analogous test cases for other registrar systems.

2.1 Contact

2.1.1 Summary

<table>
<thead>
<tr>
<th>Operation</th>
<th>Outcome</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create</td>
<td>success</td>
<td>Used بازار@أكان.بازار as email address</td>
</tr>
<tr>
<td>Info</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Update</td>
<td>failure</td>
<td>Used δοκιμή@テスト.テストテスト as email address; request is rejected with reason “not a valid email address”</td>
</tr>
<tr>
<td>Update (adjusted)</td>
<td>success</td>
<td>Used 测试@测试.测试 as email address</td>
</tr>
</tbody>
</table>

For the Contact test cases only the email address was of concern. Any other data, which may contain the full range of unicode characters (e.g. internationalised address data), is ignored in the context of Universal Acceptance.

It should be tested that an EAI is accepted when storing and updating contacts and that the email address is correctly returned when inquiring the created contact.
Testing the GatewayNG registrar system resulted in two problems:

- The email address <δοκιμή@テスト.パリス> was not accepted. The cause being a faulty validation of the javax.mail library used to validate email addresses. See Section “Third-Party Library Usage” of the main document for details.

Starting with Release 1.59 of the GatewayNG the email validation has been adjusted and the above email address is considered valid and accepted.

- The domain name parts are not normalised to U-label notation, instead they are only normalised to lower-case and otherwise stored and displayed as given, resulting in searches (in the web interface) not finding contacts using the A-label notation when an email address was given in U-label notation.

2.1.2 Detailed Test Descriptions

2.1.2.1 Create

The formal definition of the example used in this test can be found in [GWContactCreate].

Request:

payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657264334464
core.member.id: CORE-1
request.type: contact.create
registry.id: doticann
contact.authinfo: 3-YZCHErXLKilj3
contact.email: بازار@اکن.بازار
contact.i15d.address.city: پاریس
contact.i15d.address.countrycode: AX
contact.i15d.address.postalcode: پاریس
contact.i15d.address.state: پاریس
contact.i15d.address.street.1: پاریس
contact.i15d.name: پاریس
contact.i15d.organization: پاریس

Expected result code 10000: success

Response:

contact.id: C16-T
payload.version: 2.0
provider.chain.1.spec: {MEM-CORE1}{MEM-qrZmxW}
provider.chain.1.type: default
2.1.2.2 Info

The formal definition of the example used in this test can be found in [GWContactInquire].

Request:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657265333394
core.member.id: CORE-1
request.type: contact.inquire
registry.id: doticann
contact.id: C16-T

Expected result code 10000: success
Check that email has the same value as in the create request: success
Response:
contact.authinfo: 3-YZCHeXLKilj3
contact.datapolicy: restrictive
contact.email: بازار@بازار
contact.email.verification.status: unknown
contact.i15d.address.city: பரிட்சை
contact.i15d.address.countrycode: AX
contact.i15d.address.postalcode: பரிட்சை
contact.i15d.address.state: பரிட்சை
contact.i15d.address.street.1: பரிட்சை
contact.i15d.name: பரிட்சை
contact.i15d.organization: பரிட்சை
contact.id: C16-T
contact.status: ok
creation.date: 2022-07-08T07:16:04.456Z
creator.client.id: reg-990386
creator.core.member.id: CORE-1
payload.version: 2.0
provider.chain.1.spec: {MEM-CORE1}{MEM-qrZmxW}
provider.chain.1.type: default
registry.transaction.id: 1657265280103-885
response.type: contact.inquire
result.code: 10000
result.msg: Command completed successfully
sponsor.client.id: reg-990386
sponsor.core.member.id: CORE-1
transaction.id: GWWeb-michael.bauland-1657265333394
2.1.2.3 Update
The formal definition of the example used in this test can be found in [GWContactModify].

Request:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657265333394
core.member.id: CORE-1
request.type: contact.modify
registry.id: doticann
contact.id: C16-T
contact.email: δοκιμή@テスト.テスト.
contact.i15d.address.city: பரிட்சை
contact.i15d.address.countrycode: AX
contact.i15d.address.postalcode: பரிட்சை
contact.i15d.address.state: பரிட்சை
contact.i15d.address.street.1: பரிட்சை
contact.i15d.name: பரிட்சை

Expected result code 10000: fail
Response:
payload.version: 2.0
response.type: contact.modify
result.1.code: 20005
result.1.error.1: not a valid email address
result.1.error.2: δοκιμή@テスト.テスト.
result.1.key.1: contact.email
result.1.msg: Parameter value syntax error
result.code: 20101
result.msg: Payload specification violation
transaction.id: GWWeb-michael.bauland-1657265333394
2.1.2.4 Update (adjusted)

The formal definition of the example used in this test can be found in [GWContactModify].

Alter test to use different email address: 测试@测试.测试

Request:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657265333394
core.member.id: CORE-1
request.type: contact.modify
registry.id: doticann
contact.id: C16-T
contact.email: 测试@测试.测试

contact.i15d.address.city: பரிட்சை
contact.i15d.address.countrycode: AX
contact.i15d.address.postalcode: பரிட்சை
contact.i15d.address.state: பரிட்சை
contact.i15d.address.street.1: பரிட்சை
contact.i15d.name: பரிட்சை

Expected result code 10000: success

Response:
payload.version: 2.0
response.type: contact.modify
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657265333394
2.2 Host
The formal definition of the example used in this test can be found in [GWHostCreate].

2.2.1 Summary

<table>
<thead>
<tr>
<th>Operation</th>
<th>Outcome</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create (U-label)</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Create (A-label)</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Info</td>
<td>success</td>
<td>The returned host is in A-label and U-label notation.</td>
</tr>
<tr>
<td>Update</td>
<td>n/a</td>
<td>No renaming of hosts is allowed.</td>
</tr>
</tbody>
</table>

Contrary to EPP the Payload protocol has no restriction to A-labels for host names. It therefore should be tested that host names are accepted both as A-labels and U-labels.

Testing the GatewayNG registrar system showed full support of arbitrary A-labels and U-labels.
2.2.2 Detailed Test Descriptions

2.2.2.1 Create (U-label)

The formal definition of the example used in this test can be found in [GWHostCreate].

Request:
```
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657270363719
core.member.id: CORE-1
request.type: host.create
registry.id: doticann
host.name: إختبار.آزمایشی
```

Expected result code 10000: success
The A-label version of the host is correct: success

Result:
```
host.id: Hkeln11-ICANN
host.name: xn--kgbechtv.xn--1-omcp7b14hw8bba.xn--11b5bs3a9aj6g
host.name.il5d: إختبار.آزمایشی
payload.version: 2.0
provider.chain.1.spec: {MEM-CORE1}{MEM-qrZmxW}
provider.chain.1.type: default
registry.transaction.id: 1657270419546-909
response.type: host.create
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657270363719
2.2.2 Create (A-label)

The formal definition of the example used in this test can be found in [GWHostCreate].

Request:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657270363719
core.member.id: CORE-1
request.type: host.create
registry.id: doticann
host.name: xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g

Expected result code 10000: success
The U-label version of the host is correct: success

Result:
host.id: Hbsiw12-ICANN
host.name: xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g
host.name.i15d: اختبار.اخرص
payload.version: 2.0
provider.chain.1.spec: {MEM-CORE1}{MEM-qrZmxW}
provider.chain.1.type: default
registry.transaction.id: 1657270522524-913
response.type: host.create
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657270363719
2.2.2.3 Info
The formal definition of the example used in this test can be found in [GWHostInquire].

Request 1:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657270363719
core.member.id: CORE-1
request.type: host.inquire
registry.id: doticann
host.id: Hbsiw12-ICANN

Expected result code 10000: success
The A-label version is correct: success
The U-label version is correct: success
Result:
creation.date: 2022-07-08T08:55:22.524Z
creator.client.id: reg-990386
creator.core.member.id: CORE-1
host.id: Hbsiw12-ICANN
host.name: xn--kgbechtv.xn--1-omcp7b14hw8bba.xn--11b5bs3a9aj6g
host.name.i15d: اختبار.ارماني.پنگ
host.status: ok
payload.version: 2.0
provider.chain.1.spec: {MEM-CORE1}{MEM-qrZmxW}
provider.chain.1.type: default
registry.transaction.id: 1657271307006-919
response.type: host.inquire
result.code: 10000
result.msg: Command completed successfully
sponsor.client.id: reg-990386
sponsor.core.member.id: CORE-1
transaction.id: GWWeb-michael.bauland-1657270363719

2.2.2.4 Update
As modifying hosts does not allow changing the domain name of the host, there is nothing to test here.
2.3 Domain

2.3.1 Summary

<table>
<thead>
<tr>
<th>Operation</th>
<th>Outcome</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create (U-label)</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Create (A-label)</td>
<td>success</td>
<td></td>
</tr>
<tr>
<td>Info</td>
<td>success</td>
<td>The returned domain is in A-label and U-label notation.</td>
</tr>
<tr>
<td>Update (setting variants)</td>
<td>success</td>
<td>Submitting variants in A-label and U-label notation</td>
</tr>
<tr>
<td>Check (variant)</td>
<td>success</td>
<td>Checking a non-existing domain, which is a variant of an existing domain reports the domain as blocked</td>
</tr>
<tr>
<td>Check (non-variant)</td>
<td>success</td>
<td>Checking a non-existing domain, which is not a variant of an existing domain reports the domain as available</td>
</tr>
<tr>
<td>Transfer</td>
<td>success</td>
<td>Starting a transfer using the A-label and U-label notation</td>
</tr>
</tbody>
</table>

Contrary to EPP the Payload protocol has no restriction to A-labels for host names. It therefore should be tested that domain names are accepted both as A-labels and U-labels. The support of variants and IDN script/language values is also tested.

Testing the GatewayNG registrar system showed full support of arbitrary A-labels and U-labels.
2.3.2 Detailed Test Descriptions
The formal definition of the example used in this test can be found in [GWDomainCreate].

2.3.2.1 Create (U-label)

Request:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.create
registry.id: doticann
domain.name: këst.テスト
domain.name.language: fr
domain.authinfo: abcdefghijk
contact.1.id: REG-CEUOBXDWZJF2
contact.1.type: admin
contact.2.id: REG-CEUOBXDWZJF2
contact.2.type: registrant
contact.3.id: REG-CEUOBXDWZJF2
contact.3.type: tech
contact.4.id: REG-CEUOBXDWZJF2
contact.4.type: billing
period.unit: y
period.value: 1

Expected result code 10000: success
The A-label version of the domain is correct: success
Result:
account.1.balance: -433.80
account.1.change: -7.80
account.1.currency: USD
domain.id: Dzdko9-ICANN
domain.name: xn--kst-jma.xn--zckzar
domain.name.ill5d: këst.テスト
domain.name.ill5d: këst.テスト
expiration.date: 2023-07-14T10:34:26.802Z
payload.version: 2.0
provider.chain.1.spec: {MEM-CORE1}{MEM-qrZmxW}
provider.chain.1.type: default
registrant.verification.started: true
registry.transaction.id: 1657794866802-2819
response.type: domain.create
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657793947373
2.3.2.2 Create (A-label)

The formal definition of the example used in this test can be found in [GWDomainCreate].

Request:

```plaintext
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.create
registry.id: doticann
domain.name: xn--kst-jma.xn--zckzah
domain.name.language: fr
domain.authinfo: abcdefghijk
contact.1.id: REG-CEUOBXDWZJF2
contact.1.type: admin
contact.2.id: REG-CEUOBXDWZJF2
contact.2.type: registrant
contact.3.id: REG-CEUOBXDWZJF2
contact.3.type: tech
contact.4.id: REG-CEUOBXDWZJF2
contact.4.type: billing
period.unit: y
period.value: 1

Expected result code 10000: success
The U-label version of the domain is correct: success

Result:

```plaintext
account.1.balance: -433.80
account.1.change: -7.80
account.1.currency: USD
domain.id: Dmjag10-ICANN
domain.name: xn--kst-jma.xn--zckzah
domain.name.i15d: këst.テスト
expiration.date: 2023-07-14T11:21:18.260Z
payload.version: 2.0
provider.chain.1.spec: {MEM-CORE1}{MEM-qrZmxW}
provider.chain.1.type: default
registrant.verification.started: true
registry.transaction.id: 1657797678260-2837
response.type: domain.create
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657793947373
```
2.3.2.3 Info

The formal definition of the example used in this test can be found in [GWDomainInquire].
In the first request, the domain is written in A-label notation, in the second request, the domain is written in U-label notation.

Request 1:
```
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.inquire
registry.id: doticann
domain.name: xn--kst-jma.xn--zckzah
```

Request 2:
```
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.inquire
registry.id: doticann
domain.name: këst.テスト
```

Expected result code 10000: success
Both requests return the same data: success
A-label and U-label version of the domain is correct: success

Result:
```
contact.1.id: REG-CEUOBXDWZJF2
contact.1.type: registrant
contact.2.id: REG-CEUOBXDWZJF2
contact.2.type: tech
contact.3.id: REG-CEUOBXDWZJF2
contact.3.type: admin
contact.4.id: REG-CEUOBXDWZJF2
contact.4.type: billing
creation.date: 2022-07-14T11:21:18.260Z
creator.client.id: reg-990386
creator.core.member.id: CORE-1
domain.authinfo: abcdefghijk
domain.id: Dmjag10-ICANN
domain.name: xn--kst-jma.xn--zckzah
domain.name.i15d: këst.テスト
domain.name.language: fr
domain.status: inactive
expiration.date: 2023-07-14T11:21:18.260Z
launch.phase: open
2.3.2.4 Update (setting variants)

The formal definition of the example used in this test can be found in [GWDomainModify]. One variant is set in A-label the other in U-label format.

Request:

payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.modify
registry.id: doticann
domain.name: këst.テスト
domain.variant.1.name: kest.xn--zckzah
domain.variant.2.name: xn--kst-bma.xn--zckzah
domain.name.language: fr
update: idn

Expected result code 10000: success

Result:

payload.version: 2.0
registry.transaction.id: 1657798981194-2850
response.type: domain.modify
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657793947373
2.3.2.5 Check (variant)

The formal definition of the example used in this test can be found in [GWDomainCheck].

Request:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.check
registry.id: doticann
domain.1.name: xn--kst-bma.xn--zckzah
domain.1.name.language: fr

Expected result code 10000: success
Expected availability “false”: success

Result:
domain.1.available: false
domain.1.name: xn--kst-bma.xn--zckzah
domain.1.name.ill5d: kést.テスト
domain.1.reason: Blocked by variant
payload.version: 2.0
registry.transaction.id: 1657799223966-2852
response.type: domain.check
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657793947373
2.3.2.6 Check (non-variant)

The formal definition of the example used in this test can be found in [GWDomainCheck].

Request:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.check
registry.id: doticann
domain.1.name: mést.テスト
domain.1.name.language: fr

Expected result code 10000: success
Expected availability “true”: success

Result:
domain.1.available: true
domain.1.name: xn--mst-bma.xn--zckzah
domain.1.name.i15d: mést.テスト
payload.version: 2.0
registry.transaction.id: 1657799539851-2859
response.type: domain.check
result.code: 10000
result.msg: Command completed successfully
transaction.id: GWWeb-michael.bauland-1657793947373
2.3.2.7 Transfer

The formal definition of the example used in this test can be found in [GWDomainTransfer]. The domain is transferred both using the A-label and using the U-label notation.

Request 1:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.transfer.request
registry.id: doticann
domain.name: xn--rst-bma.xn--zckzah
domain.authinfo: abcdefg

Request 2:
payload.version: 2.0
transaction.id: GWWeb-michael.bauland-1657793947373
core.member.id: CORE-1
request.type: domain.transfer.request
registry.id: doticann
domain.name: rést.テスト
domain.authinfo: abcdefg

Expected result code 10100: success

Result:
account.1.balance: -439.00
account.1.change: -5.20
account.1.currency: USD
payload.version: 2.0
registry.transaction.id: 1657799953537-2862
response.type: domain.transfer.request
result.code: 10100
result.msg: Command accepted
transaction.id: GWWeb-michael.bauland-1657793947373
3 Testing the Authoritative DNS Server - G1

The test cases within this section reflect the UA-readiness gate G1 as visualized in Figure 2: Domain Registrar High-Level System Architecture in the main document. While running an authoritative DNS server for their domains is not mandatory for registrars, most registrars also offer this service for their customers. For this test case a domain needs to be created and “activated” at the registrars, i.e., it needs to be assigned name servers and the registrar needs to create a zone for the domain on those name servers.

3.1 DNS Query

For this test we will be using the domain këst.テスト as created in the Payload Domain test case. Its A-label notation is xn--kst-jma.xn--zckzah. To test the DNS server the open source tool “dig” is used.

3.1.1 Detailed Test Description

Query the domain name’s SOA record from the domain’s configured authoritative name server ns1.sandbox.irondns.net:

```
dig @ns1.sandbox.irondns.net xn--kst-jma.xn--zckzah SOA
```

Expected outcome:
- Status: NOERROR: success
- SOA record is returned in Answer Section: success
Result:

; <<>> DiG 9.10.3-P4-Ubuntu <<>> @ns1.sandbox.irondns.net xn--kst-jma.xn--zckzah soa
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; -->>HEADER<<-- opcode: QUERY, status: NOERROR, id: 22312
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
xn--kst-jma.xn--zckzah. IN SOA

;; ANSWER SECTION:
xn--kst-jma.xn--zckzah. 86400 IN SOA ns1.sandbox.irondns.net secretariat.corenic.org. 2022071501 10800 10800 604800 3600

;; AUTHORITY SECTION:
xn--kst-jma.xn--zckzah. 86400 IN NS ns1.sandbox.irondns.net.

;; Query time: 4 msec
;; SERVER: 2a01:5b0:0:126::12#53(2a01:5b0:0:126::12)
;; WHEN: Fri Jul 15 08:56:34 CEST 2022
;; MSG SIZE  rcvd: 148

;; MSG SIZE  rcvd: 87
4 Testing the Port 43 Whois Interface - G2

The test cases within this section reflect the UA-readiness gate G2 as visualized in Figure 2: Domain Registrar High-Level System Architecture in the main document. For these test cases a domain, a contact, and a host need to be created.

4.1 Contact

4.1.1 Detailed Test Description

Query an existing contact via its handle:

whois -h whois-ua-test-rr-icann.knipp.de contact C10-T

Expected outcome:
- Email address is correctly returned: success

Result:
Registry Contact ID: C10-T
Contact Name: பரிட்சை
Contact Organization:
Contact Street: பரிட்சை
Contact City: பரிட்சை
Contact State/Province:
Contact Postal Code: பரிட்சை
Contact Country:
Contact Phone:
Contact Phone Ext:
Contact Fax:
Contact Fax Ext:
Contact Email: example@xn--0zwm56d.xn--0zwm56d

>>> Last update of Whois database: 2022-07-15T07:15:00.306Z <<<
4.2 Host

4.2.1 Detailed Test Description

Query an existing host via its domain name:

`whois -h whois-ua-test-rr-icann.knipp.de host xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g`

Expected outcome:
- Host is found: success
- Host’s U-label is correctly returned: success

Result:

Server Name: xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g
Registrar: COREhub, S.R.L.
Registrar WHOIS Server: whois.corenic.net
Registrar URL: http://www.corenic.net
Internationalized Server Name: آزمایشی.إختبار.ازmaalشي

>>> Last update of Whois database: 2022-07-18T04:51:22.10Z <<<
4.3 Domain

4.3.1 Detailed Test Description

Query an existing domain via its domain name:

```bash
whois -h whois-ua-test-rr-icann.knipp.de xn--kst-jma.xn--zckzah
```

Expected outcome:
- Domain is found: success
- Domain’s U-label is correctly returned: success
- Registrar email address is correctly returned: success
- Contact email addresses are correctly returned: success

Result:

**Domain Name:** xn--kst-jma.xn--zckzah

**Internationalized Domain Name:** kést.テスト

Registry Domain ID: Dmjag10-ICANN
Registrar WHOIS Server: whois.corenic.net
Registrar URL: http://www.corenic.net
Updated Date: 2022-07-14T11:43:01.93Z
Creation Date: 2022-07-14T11:21:18.260Z
Registrar Registration Expiration Date: 2023-07-14T11:21:18.260Z
Registrar: COREhub, S.R.L.
Registrar IANA ID: 15
Registrar Abuse Contact Email: example@xn--0zwm56d.xn--0zwm56d
Registrar Abuse Contact Phone: +34.343434
Reseller: CORE-1 (UA Test Client)
Domain Status: ok https://icann.org/epp#ok
Name Server:
DNSSEC: unsigned
IDN Tag: fr
URL of the ICANN WHOIS Data Problem Reporting System:
http://wdprs.internic.net/

5 Testing the RDAP Interface - G3

The test cases within this section reflect the UA-readiness gate G3 as visualized in Figure 2: Domain Registrar High-Level System Architecture in the main document. For these test cases a domain, a contact, and a host need to be created.

5.1 Contact

5.1.1 Detailed Test Description

Query an existing contact via its handle:

https://whois-ua-test-rr-icann.knipp.de/rdap/entity/C10-T

Expected outcome:

- Email address is correctly returned: success

Result:

```json
{
    "rdapConformance": [
        "rdap_level_0",
        "icann_rdap_response_profile_0",
        "icann_rdap_technical_implementation_guide_0"
    ],
    "notices": [
        {
            "title": "Terms of Service",
            "description": [
                "cut"
            ],
            "links": [
                {
                    "href": "https://whois-ua-test-rr-icann.knipp.de/rdap-tas"
                }
            ]
        }
    ],
    "remarks": [
        {
            "title": "REDACTED FOR PRIVACY",
            "description": [
                "Some of the data in this object has been removed"
            ],
            "type": "object redacted due to authorization"
        }
    ]
}
```

},
"objectClassName": "entity",
"handle": "C10-T",
"events": [
  {
    "eventAction": "last update of RDAP database",
    "eventDate": "2022-07-18T05:18:24.122Z"
  },
  {
    "eventAction": "registration",
    "eventActor": "core",
    "eventDate": "2022-07-05T06:47:37.531Z"
  },
  {
    "eventAction": "last changed",
    "eventActor": "core",
    "eventDate": "2022-07-05T08:14:07.951Z"
  }
],
"links": [
  {
    "value": "https://whois-ua-test-rr-icann.knipp.de/rdap/entity/C10-T",
    "rel": "self",
    "href": "https://whois-ua-test-rr-icann.knipp.de/rdap/entity/C10-T",
    "type": "application/rdap+json"
  }
],
"port43": "whois.corenic.net",
"vcardArray": [
  "vcard",
  [
    "version", { }, "text", "4.0" ],
  [ "fn", { }, "text", "பரிட்சை" ],
  [ "adr", { }, "text", [ "", "", "பரிட்சை", "பரிட்சை", "பரிட்சை", "" ] ],
  [ "email", { }, "text", "example@xn--0zwm56d.xn--0zwm56d" ]
]
5.2 Host

5.2.1 Detailed Test Description

Query an existing host via its domain name in A-label and U-label notation:

https://whois-ua-test-rr-icann.knipp.de/rdap/nameserver/xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g

Expected outcome:
- Host is found: success
- Host’s A-label is correctly returned: success
- Host’s U-label is correctly returned: success
- Both queries return the same data: success

Result:
{
    "rdapConformance": [
        "rdap_level_0", "icann_rdap_response_profile_0", "icann_rdap_technical_implementation_guide_0"
    ],
    "notices": [
        {
            "title": "Terms of Service",
            "description": [
                "cut"
            ],
            "links": [
                {
                    "href": "https://whois-ua-test-rr-icann.knipp.de/rdap-tas"
                }
            ]
        }
    ],
    "objectClassName": "nameserver",
    "handle": "Hbsiw12-ICANN",
    "events": [
        {
            "eventAction": "last update of RDAP database",
            "eventDate": "2022-07-18T05:30:25.25Z"
        }
    ]
}
"eventAction": "registration",
"eventActor": "core",
"eventDate": "2022-07-08T08:55:22.524Z"
]
"links": [
{
"value": "https://whois-ua-test-rr-icann.knipp.de/rdap/nameserver/xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g",
"rel": "self",
"href": "https://whois-ua-test-rr-icann.knipp.de/rdap/nameserver/xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g",
"type": "application/rdap+json"
}
],
"ldhName": "xn--kgbechtv.xn--1-omcp7bl4hw8bba.xn--11b5bs3a9aj6g",
"unicodeName": "آزمایشی.تست.",
"port43": "whois.corenic.net"}
5.3 Domain

5.3.1 Detailed Test Description

Query an existing domain via its domain name in A-label and U-label notation:

https://whois-ua-test-rr-icann.knipp.de/rdap/domain/xn--kst-jma.xn--zckzah

https://whois-ua-test-rr-icann.knipp.de/rdap/domain/këst.テスト

Expected outcome:
- Domain is found: success
- Domain's A-label is correctly returned: success
- Domain's U-label is correctly returned: success
- Both queries return the same data: success
- Registrar email address is correctly returned: success
- Contact email address is correctly returned: success

Result:

```json
{
  "rdapConformance": [
    "rdap_level_0", "icann_rdap_response_profile_0",
    "icann_rdap_technical_implementation_guide_0"
  ],
  "notices": [
    {
      "title": "Status Codes",
      "description": [
        "For more information on domain status codes, please visit https://icann.org/epp"
      ],
      "links": [
        {
          "href": "https://icann.org/epp"
        }
      ]
    },
    {
      "title": "RDDS Inaccuracy Complaint Form",
      "description": [
        "URL of the ICANN RDDS Inaccuracy Complaint Form: https://icann.org/wicf"
      ]
    }
  ]
}
```
"links": [
    {
        "href": "https://icann.org/wicf"
    }
],
"title": "Terms of Service",
"description": [
    "cut"
],
"links": [
    {
        "href": "https://whois-ua-test-rr-icann.knipp.de/rdap-tas"
    }
],
"objectClassName": "domain",
"handle": "Dmjag10-ICANN",
"status": ["active"],
"events": [
    {
        "eventAction": "last update of RDAP database",
        "eventDate": "2022-07-18T05:38:25.717Z"
    },
    {
        "eventAction": "registration",
        "eventActor": "core",
        "eventDate": "2022-07-14T11:21:18.260Z"
    },
    {
        "eventAction": "last changed",
        "eventActor": "core",
        "eventDate": "2022-07-14T11:43:01.93Z"
    },
    {
        "eventAction": "expiration",
        "eventDate": "2023-07-14T11:21:18.260Z"
    }
],
"links": [
    {
        "value": "https://whois-ua-test-rr-icann.knipp.de/rdap/domain/xn-kst-jma.xn--zckzah",
        "rel": "self",
        "href": "https://whois-ua-test-rr-icann.knipp.de/rdap/domain/xn-kst-jma.xn--zckzah",
        "type": "application/rdap+json"


```json
{
  "ldhName": "xn--kst-jma.xn--zckzah",
  "unicodeName": "këst.テスト",
  "variants": [
    {
      "relation": [ "registered" ],
      "variantNames": [
        {
          "ldhName": "kest.xn--zckzah",
          "unicodeName": "kest.テスト"
        },
        {
          "ldhName": "xn--kst-bma.xn--zckzah",
          "unicodeName": "kést.テスト"
        }
      ],
      "idnTable": "fr"
    }
  ],
  "entities": [
    {
      "objectClassName": "entity",
      "handle": "REG-CEUOBXDWZJF2",
      "links": [
        {
          "value": "https://whois-ua-test-rr-icann.knipp.de/rdap/entity/REG-CEUOBXDWZJF2",
          "rel": "self",
          "href": "https://whois-ua-test-rr-icann.knipp.de/rdap/entity/REG-CEUOBXDWZJF2",
          "type": "application/rdap+json"
        }
      ],
      "roles": [ "registrant" ]
    },
    {
      "objectClassName": "entity",
      "handle": "REG-CEUOBXDWZJF2",
      "links": [
        {
          "value": "https://whois-ua-test-rr-icann.knipp.de/rdap/entity/REG-CEUOBXDWZJF2",
          "rel": "self",
          "href": "https://whois-ua-test-rr-icann.knipp.de/rdap/entity/REG-CEUOBXDWZJF2",
          "type": "application/rdap+json"
        }
      ],
      "roles": [ "registrant" ]
    }
  ]
}
```
"type": "application/rdap+json"

"roles": [ "technical", "billing", "administrative" ]

"objectClassName": "entity",
"handle": "15",
"publicIds": [
  {
    "type": "IANA Registrar ID",
    "identifier": "15"
  }
],
"roles": [ "registrar" ],
"entities": [ { "objectClassName": "entity", "handle": "not applicable", "roles": [ "abuse" ],
  "vcardArray": [ { "vcard": [ [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "Abuse Contact" ],
    [ "adr", { }, "text", [ "", "", "", "", "", "", "" ] ], [ "tel", { "type": "voice" }, "URI", "tel:+34.34343434" 
  ], [ "email", { }, "text", "example@xn--0zwm56d.xn--0zwm56d" ] ]
  } }, { "objectClassName": "entity", "roles": [ "administrative" ],
  "vcardArray": [ { "vcard": [ [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "COREhub, S.R.L." ],
    [ "adr", { }, "text", [ "", "", "", "", "", "", "" ] ], [ "tel", { "type": "voice" }, "URI", "tel:+34.34343434" ] ]
  ] } ]}
"objectClassName": "entity",
"roles": [ "technical" ],
"vcardArray": [
  "vcard",
  [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "COREhub, S.R.L." ],
  [ "adr", { }, "text", [ "", "", "", "", "", "", "" ] ], [ "tel", { "type": "voice" }, "URI", "tel:+34.343434" ]
],
"vcardArray": [
  "vcard",
  [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "COREhub, S.R.L." ],
  [ "adr", { }, "text", [ "", "", "", "", "", "", "" ] ]
],
"objectClassName": "entity",
"handle": "CORE-1",
"roles": [ "reseller" ],
"vcardArray": [
  "vcard", [ [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "UA Test Client" ] ]
],
"secureDNS": {
  "delegationSigned": false
},
"port43": "whois.corenic.net",
"lang": "fr"
6 Testing the Escrow Export - G10

The test case within this section reflects the UA-readiness gate G10 as visualized in Figure 2: Domain Registrar High-Level System Architecture in the main document. For this test case domains, contacts and hosts must have been created before the Escrow’s reference date.

6.1 Escrow

6.1.1 Detailed Test Description

After the respective objects have been created, the Escrow export needs to be triggered. The generated escrow file needs to contain all objects with their correct names.

Expected outcome:
- Domain (A-label) is contained: success
- Host (A-label) is contained: success
- Contact is contained with correct email address: success

Result:
Note that the Escrow generated by the GatewayNG is using the CSV format. It consists of two files. The first file contains the domain data, the second file contains details of the referenced handle. The content has been redacted to only show the tested objects.

Domain Escrow

domain-name,name-servers,expiration-date,rt-handle,ac-handle,tc-handle,bc-handle

Referenced Handles

domain-name,organization,street1,street2,street3,city,state,postal-code,country,voice,voice-ext,fax,fax-ext,email
R10-C10-T,பரிட்சை,பரிட்சை,பரிட்சை,,,பரிட்சை,பரிட்சை,பரிட்சை,Ax,,,,,example@xn
-0zwm56d.xn--0zwm56d
7 Testing Sending of Emails - G6

The test case within this section reflects the UA-readiness gate G6 as visualized in Figure 2: Domain Registrar High-Level System Architecture in the main document. For this test case a contact and a domain need to exist.

7.1 Whois Accuracy Program

7.1.1 Detailed Test Description

The contact to be used should have a not yet validated email address for which mails can be received: michael@xn--igbi7fn.xn--mgbab2bd. Assign the contact to an existing domain as registrant contact. The registrar system should then validate the contact’s email address according to the Whois Accuracy Program (WAP) as mandated by ICANN by sending a confirmation email to the address.

Expected outcome:
- The email is received: success
- The domain name of the connected domain is displayed correctly: success

Result:
Email with subject “[ACTION REQUIRED] E-mail address validation for 1 domain: këst.テスト” was received at michael@xn--igbi7fn.xn--mgbab2bd.

Adjust the test to use a contact with Unicode at the local part of the email address, i.e., grūün@knipp.de.

Expected outcome:
- The email is received: success
- The domain name of the connected domain is displayed correctly: success

Result:
Email with subject “[ACTION REQUIRED] E-mail address validation for 1 domain: këst.テスト” was received at mailbox for grūün@knipp.de.
7.2 Transfer Notifications

7.2.1 Summary
Testing the transfer notification email revealed that the emails subject did not have the correct encoding. It used ASCII instead of UTF-8, causing the subject not to be displayed correctly. Starting with Release 1.60 of the GatewayNG the encoding has been fixed and the domain name is displayed correctly in the email’s subject.

7.2.2 Detailed Test Description
Initiate a transfer of an existing domain. Once the transfer is completed a confirmation e-mail is sent to the registrant contact.

Expected outcome:
- The e-mail is received: success
- The domain is displayed correctly:
  - Subject: fail
  - Body: success

Result:
Subject line: “incoming domain transfer notification: ????”
Body (redacted): “the transfer of the domain ‘test.テスト’ (requested by you) is finished.”
Bibliography

[GWContactCreate]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.1.html
[GWContactInquire]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.13.html
[GWContactModify]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.23.html
[GWDomainCheck]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.17.html
[GWDomainCreate]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.4.html
[GWDomainInquire]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.18.html
[GWDomainModify]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.25.html
[GWDomainTransfer]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.27.html
[GHHostCreate]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.3.html
[GHHostInquire]  https://ua-test-rr-icann.knipp.de/api/docs/mp/section-2.15.html
[GWPayload]  https://ua-test-rr-icann.knipp.de/api/docs/mp/index.html
[vue]  https://vuejs.org/
[vuelidate]  https://vuelidate.js.org/