Registry System Testing

IDN Test Area Specification

Version ~~C~~D

File name: Test Area IDN.docx
Last saved: 2021-06-09

Document control

Document information and security

Made by	Responsible for fact	Responsible for document
Lennart Bonnevier	Mats Dufberg	Mats Dufberg

Security class	File name
External	Test Area IDN.docx

Revisions

Date	Version	Name	Description
2017-07-28	A	Mats Dufberg	First release version.
2017-07-28	B	Mats Dufberg	Added reference to LGR formatted IDN tables available at ICANN.
2018-07-19	C	Mats Dufberg	Update to only refer to the LGR formatted IDN tables available at ICANN (not to the IIS tables). IDNValid03 updated to explicitly require the contextual rules found in relevant reference table.
2021-06-10	D	Mauro Lozano	Update references.

LIST OF CONTENTS

1.   Introduction

1.1   Scope

This document describes the IDN Level Tests within the Registry System Testing (RST) framework. These tests are for zones that support IDN and are not applicable to zones that are ASCII only.

The RST Service Provider will verify that there is a one-to-one correspondence between the languages and/or scripts listed in Exhibit A of the TLD Registry Agreement and the IDN tables provided; that each IDN table is explicitly associated with a single script or language; that each IDN table is formatted according to RFC 4290, RFC 3743 or RFC 7940; that each listed code point is valid under the IDNA 2008 protocol; that every string of tabulated code points permissible as a registered label conforms to the IDN Guidelines: that all policy and context-dependent requirements of IDNA 2008 and the Guidelines are clearly stated and enforced in the registry; that a complete list has been provided of the language or script tags for the IDN EPP Extension, if such is required (e.g. 'fr' for a French language IDN table, or 'cyrl' for a Cyrillic script IDN table); that all policies and rules associated with the management of variant relationships among tabulated code points are documented and enforced; that all contextual rules are documented and enforced; that a specimen EPP transaction has been provided illustrating how to register a new IDN label, including a language or script tag if they are used.

1.2   References

1.2.1   External

- ICANN gTLD Applicant Guidebook, Version 2012-06-04
- ICANN Guidelines for the Implementation of Internationalized Domain Names ("IDN Guidelines")
- IEEE 829-2008
- RFC 3743, Guidelines for Internationalized Domain Names (IDN) Registration and Administration for Chinese, Japanese, and Korean
- RFC 4290, Suggested Practices for Registration of Internationalized Domain Names (IDN)
- RFCs 5890 through 5894, jointly referred to as IDNA2008
- RFC 7940, Representing Label Generation Rulesets Using XML
- The Unicode Standard ("Unicode"), all versions 5.2 and higher

1.2.2   Internal

- Pre-Delegation Testing, ~~Statement of Work~~
- ~~Pre-Delegation Testing,~~ Master Test Plan

1.2.3 Document Hierarchy
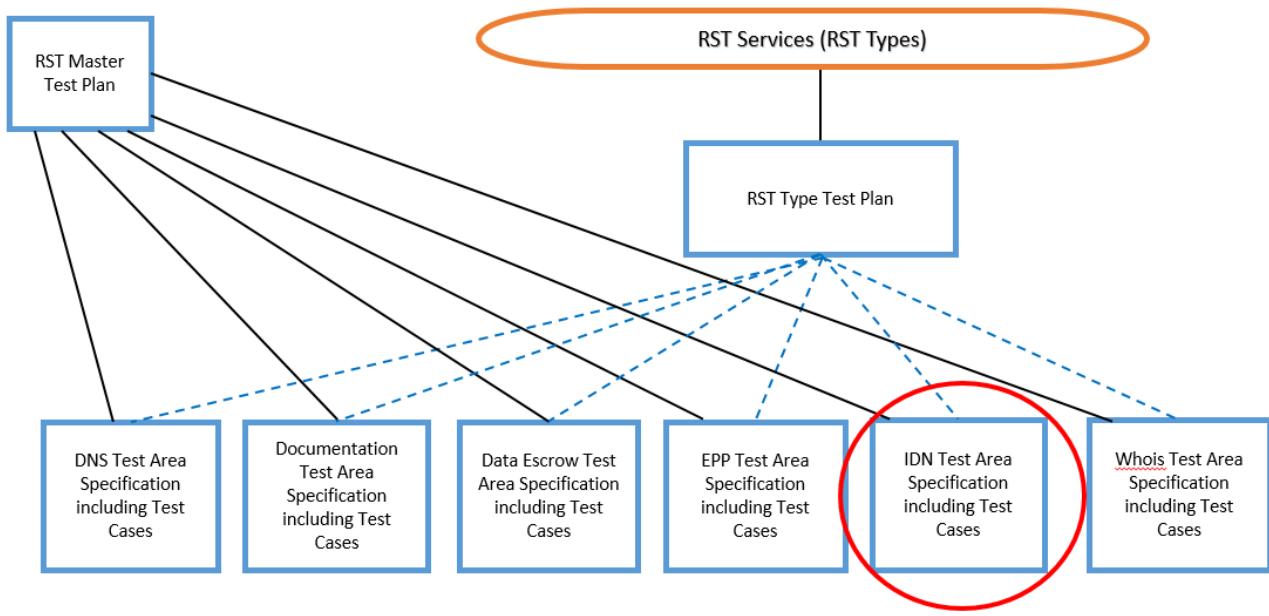
This document is one of many Test Area Specifications for RST (circled in red in the above graphic). It defines the Test Cases for its Test Area.

1.3 Earlier documents

This document replaces, in contents, the following documents that were part of PDT (Pre-Delegation Testing):

- Pre-Delegation Testing: IDN Test Plan (version I)
- Pre-Delegation Testing: IDN Test Cases (version O)

1.4 Level in the overall sequence

The IDN Test Area and the associated Test Cases can be run in parallel with the other Test Areas.

2. Test Requirements

2.1 Test items and their identifiers

2.1.1 Statement of Work

IDN attributes to be tested that are specified in the Statement of Work:

[R19] Verify the complete IDN tables(s) used by the registry system for compliance with the Guidelines for the Implementation of Internationalized Domain Names.

[R20] Verify the format is such as the model format described in Section 5 of RFC 4290. The format used by RFC 3743 is an acceptable alternative. For variant generation algorithms that are more complex (such as those with contextual rules) and cannot be expressed using these table formats, an exception handling process, in which domains are to be tested manually, should be implemented.

IDN attributes to be tested that are based on the specification in Exhibit A of the Registry Agreement for the TLD in question. Those covers:

[RA1] Lists what languages and/or scripts, if any, that may be supported through IDN for the TLD.

[RA2] Specifies if that TLD may have variant management or not.

[RA3] Specifies if that TLD may activate variants and any limitations for that.

Support for LGR IDN Table format has been added by ICANN 2017-07-31:

[ICANN1] LGR IDN Table format as specified by RFC 7940 is the preferred IDN table format.

The IDN Guidelines cited in [R19] make reference to the IDNA protocol and both documents make reference to the Unicode Standard. The specific evaluative criteria extracted from them are included in the descriptions of the individual test cases.

Test cases that verify the status of individual code points and the way in which they can be strung together to form a label, are evaluated algorithmically and provide binary results. Test cases that verify the aggregate tabular format may be evaluated empirically. This is discussed further in section 2.3, below.

2.2 Features to be tested

The following features will be tested:

- IDN table format
- IDN code point validity
- IDN contextual rule enforcement
- IDN variant code point processing
- IDN label script integrity
- Basic IDN compliance
- Compliance of variant management

2.3 Approach

2.3.1 IDN table code point selection and presentation format

The reference documents prescribed for Test Area IDN require policy declarations that are not amenable to algorithmic evaluation. A prime example of this is the IDNA requirement that a registry makes a reasoned determination of the subset of protocol-valid code points that it will support, and articulates the policies that determine how this repertoire may appear in a second-level label. Protocol validity does not, in itself, provide adequate justification for the inclusion of a code point in an IDN table.

The IDN Guidelines express further constraints on the repertoire that may appear in a single label with particular reference to Unicode Standard Annex #24, itself introducing further constraints in narrative format. The response to requirements such as these will frequently, if not invariably, also have a narrative component that can only be reviewed empirically.

The IDN tests will make a binary determination of the presence or absence of a required algorithmic or narrative policy declaration. The resulting report will, however, only comment on the substantive detail of that statement if there is reason for concern about it having negative consequence for the stability of the IDN space. This includes both infrastructural considerations and the interests of the user community. Such concerns may be noted in the evaluation report but weighted as PASS/FAIL criteria only if there is compelling reason for doing so.

2.3.2 IDN variant code point processing

The variant generation algorithms referenced in R20 are extrinsic to the code point repertoire they produce and therefore not attributes of the IDN tables under evaluation. Although RFC 4290 provides for the indication of variant relationships in an IDN table, it does not even mention variant generation algorithms. The alternative RFC 3743 does specify a format for presenting such. The RFC 7940 format (LGR), on the other hand, makes it possible to explicitly define even complex variant rules. It is therefore the preferred format for IDN tables with variant rules.

In any case, the narrative component of a text IDN table (RFC 4290 and RFC 3743) must be parsed manually to extract information needed during the verification of that IDN table. In a RFC 7940 IDN table all information can be, and should be, encoded in the formal rules.

This will involve a manual determination that the system being tested processes the variant relationships indicated in the IDN table according to the rules declared for them, with no direct concern for the manner in which that processing is implemented on the registry side, beyond the possible verification of its functionality.

The LGR format defined in RFC 7940 is a more precise format that permits machine-parsable full definition of variant rules.

2.4 Item pass/fail criteria

An IDN table will be rejected if it contains a code point that is disallowed by the IDNA 2008 protocol when its algorithms are applied to the Unicode Standard. The IDN table will also be rejected if it includes any code point requiring a contextual rule given in the protocol, if this rule is not found in the registry-specific policy statement or encoded in the LGR (RFC 7940) IDN table. Further, the inclusion of code points with the special Unicode script property values COMMON or INHERITED must respect the constraint on script mixing stated in the IDN Guidelines as well as mixing of code points with different explicit Unicode script property values must be well-motivated, e.g. in-line with common language usage.

The reference documents for the format of an IDN text table (RFC 4290 or 3743) afford latitude in the way it can be structured. If a text IDN table conforms to either of the permitted alternatives (RFC 4290 or RFC 3743), it will pass. If it does not conform to either, it will fail. If it is a synthesis of permissible options without strictly adhering to any single one of them, note will be made of that fact and a warning will appear in the level test report.

If an LGR IDN table is submitted it must match the requirements in RFC 7940 and additional requirements defined in this document.

Discrepancies between policies for managing variant relationships and the tabulated code points will be noted explicitly for each case where they are detected.

2.5 Suspension criteria and resumption requirements

If an IDN table does not include the data necessary to conduct the entire suite of tests listed in section 3, below, the test will be suspended. If requisite statements of rules and policies have not been provided, the test will also be suspended. The Registry Operator will be advised about the reasons for this suspension and the test will be resumed when the requested supplementary information has been provided or other remedial action has been taken.

2.6 Test deliverables

The IDN test level will produce:
- Level Test Logs (LTL)
- Anomaly Report (AR) in case of error
- Level Test Report (LTR)

3.   Test Traceability Matrix

This table associates each individual test case with the SoW requirement on which it is based.

Test ID	Description	Requirement Point
IDNvalid00	**IDN documentation validation** Verification of submitted IDN tables, EPP extensions, and policy statements.	R19, R20, RA1, ICANN1
IDNvalid01	**IDN table validation** Verify that the format of each submitted IDN table conforms to RFC 4290, RFC 3743 or RFC 7940.	R20, ICANN1
IDNvalid02	**IDNA code point validation** Verify that each tabulated code point is PROTOCOL VALID or has the status CONTEXTUAL RULE REQUIRED, as defined in RFC 5892 when its algorithms are applied to the Unicode Standard.	R19, ICANN1
IDNvalid03	**IDNA Context Rule validation** Verify that the contextual rules stated in RFC 5892 are included for all code points with the status CONTEXTUAL RULE REQUIRED, that the Bidi rules stated in RFC 5893 are followed for all RTL labels, that combining marks can never start a label (RFC 5891), that Modifier Letters have necessary contextual rules, that strings are in Normalization Form C and that restrictions stated by the Unicode standard are followed.	R19, R20, ICANN1
IDNvalid04	**IDN script validation** Verify that the code point array in a single IDN table is restricted to a single explicit script property value as defined in the Unicode Standard Annex #24 and that code points with the special script property values COMMON and INHERITED are reasonably associated with the designated script.	R19, ICANN1
IDNvalid05	**IDN script-mixing rule validation** Verify that an IDN table including code points with more than one script property value is associated with rules that enforce the constraints on script mixing specified in the IDN Guidelines.	R19, ICANN1
IDNvalid06	**IDN language validation** If rules associated with an IDN table are based on a language to which they apply, verify that this association is consistent with the script-based constraints in the preceding test cases, and that the code point repertoire is consistent with the established orthographic practice of the designated language, with particular regard to the comingled use of multiple scripts in individual labels.	R19, ICANN1
IDNvalid07	**IDN variant code point validation** Verify that variant relationships between listed code points are sufficiently covered by the rules given for their processing.	R20, ICANN1
IDNvalid08	(Moved to IDNvalid11 to keep test cases in a logical order.)	

Test ID	Description	Requirement Point
IDNvalid09	**Variant management** Verify that the policy of variant management of the TLD is compliant with the specification in Exhibit A.	RA2, RA3
IDNvalid10	**Basic IDN compliance** Verify that the registry system of the TLD does not accept to register any string with HYPHEN in position three and four unless that string is a valid IDN A-label.	R19
IDNvalid11	**IDN online registry response verification** Verify that test strings needed for the preceding tests are correctly processed by the online registry.	R19, R20, ICANN1
IDNvalid12	**Asymmetrical and intransitive variant rule verification** Verify that any asymmetrical or intransitive variant rules do not create security or stability issues.	ICANN
IDNvalid13	**Pre-composed vs. decomposed character equivalence verification** Verifies that cases where the same character can be represented either by a single code point (pre-composed glyph) or by a sequence of code points (decomposed glyph) are correctly handled.	

4.   Test management

The goal of this document is to describe the test cases and how the new gTLDs are tested. This is just a part of a larger project and defining test management is not part of this subproject. However, some information can be found in the Master Test Plan.

5. Testing context

5.1 Test classes and overall test conditions

The IDN table format, the listed code points, the labels that can be assembled using them, and the associated rules and policies are validated against the requirements of the given standards and guidelines.

The tests have two basic elements. The first is the offline review of an IDN table and the associated policy statements. The second is determining the registry response to the attempted registration of labels constructed specifically to verify that code points and strings not permitted for registration are rejected, and those that are permitted are accepted, with particular attention to contextual constraints imposed in the reference documents and the additional normative instruments they invoke. The online test components are aggregated into a single test, IDNvalid11.

The tests are supported algorithmically to the extent possible. However, variation in the tabulation of nominally identical code point repertoires and the substance and format of the associated policy statements, necessitates a significant amount of manual testing.

Some tests require the generation of test labels. This will initially be part of the case-by-case action of the test officers but will gradually result in a library of generally applicable test labels. In any further case where a test label needs to be custom designed, it will be added to that repository for reuse where appropriate in subsequent testing.

5.2 Notation for description

Each IDN test case is described under a separate heading, below. The test procedures are described with the test case to which they apply.

6. Test Case IDNvalid00: IDN documentation validation

6.1 Test case identifier

IDNvalid00

6.2 Objective

This test verifies that the IDN tables and documents listed in Exhibit A of the TLD Registry Agreement have all been submitted to PDT for testing and that no submitted IDN tables or documents are unlisted in the Registry Agreement.

6.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TableList	A list of all script and language IDN tables cited in the Exhibit A of the Registry Agreement.	File, Registry Agreement.
PolicyStatement	The IDN policies declared by the registry in section 4 of the IDN Self-Certification Document, corresponding to TableList.	File, IDN Self-Certification Document.
EPPtags	A list of all IDN EPP extensions, as well as language and script tags, needed for the registration of an IDN label.	File
TestTable	The IDN table under scrutiny.	File or files.

6.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable determination. If this test case ends with "not applicable", then no further test cases will be performed.

6.5 Environmental needs

- Basic desktop.

6.6 Special procedural requirements

None.

6.7 Intercase dependencies

IDNvalid01 to extract "language" element from LGR IDN table, if any.

6.8 Ordered description of steps to be taken to execute the test case

1. Determine whether the registry is authorized to support IDNs based on Exhibit A of the Registry Agreement, and if it is not, abort the remaining sequence of IDN tests.
2. Verify that every submitted TestTable corresponds to an item in TableList and that every item in TableList corresponds to a submitted TestTable.
3. Verify that the PolicyStatement in section 4 of the IDN Self-Certification Document correlates to TestTable(s) in a manner that unambiguously indicates:
   a. How requests for registration of IDN labels will be processed.
   b. How the Registry handles comingling of scripts.

      c. How the Registry handles variants.

      d. How the Registry handles contextual rules.

4. For the IDN tables listed in the response to Section 1 of the IDN Self-Certification Document, verify that the corresponding IDN tables are listed in Section 3 of that document, that every element of EPPtags corresponds to a specific TestTable and that there are no orphaned extensions or IDN tables.

5. Verify that all IDN tables have been submitted as TXT or XML (LGR) files and that any which include non-ASCII text are encoded in Unicode UTF-8.

6. For each IDN table in LGR format (RFC 7940), verify that the language or script stated in the "language" element matches the intended language and/or script from TableList.

Criteria for N/A:

- Exhibit A of the Registry Agreement does not declare support for IDN labels (Step 1).

Criteria for PASS:

- Each script or language listed in Exhibit A corresponds to a submitted IDN table and no submitted IDN table is unlisted (Step 2),
- PolicyStatement unambiguously indicates how IDN labels are processed (Step 3a-3d),
- all IDN tables are listed in the IDN Self-Certification Document Section 1 and 3; that all elements of EPPtags correspond to specific IDN tables; that there are no orphaned extensions or IDN tables (Step 4),
- all IDN tables are submitted as TXT in UTF-8 (Step 5).

Criteria for FAIL:

- The conditions in Steps 2-5 are not met or if part of the information is unclear or missing.

If this test fails, further testing will be suspended pending remedial action. If this is not undertaken within the prescribed time, the failure will be confirmed, none of the subsequent tests will be conducted, and all will fail by default.

7.   Test Case IDNvalid01: IDN table validation

7.1   Test case identifier

IDNvalid01

7.2   Objective

This test verifies that the format of an IDN table either conforms to RFC 4290, RFC 3743, or RFC 7940. The test is repeated for all IDN tables.

7.3   Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
LocalTableFormat	Describes the IDN table format in Section 2 of the IDN Self-Certification Document if it does not comply with either of the reference RFCs.	File, IDN Self-Certification Document
LocalTableJustification	Verifiable warrant in Section 2 of the IDN Self-Certification Document for using a local format instead of either of the reference RFCs.	File, IDN Self-Certification Document

None of the reference RFCs for text IDN table formats (RFC 4290 and RFC 3743) specifies a rigorous enough format for TestTable to be automatically parsed for conformance, and there is no way to predict the details of an instance of LocalTableFormat. Manual examination of a IDN table is necessary in order to validate the format.

An IDN table in LGR format (RFC 7940) has a machine-readable format that can be automatically checked for conformance.

7.4   Outcome(s)

The response to this test will be a pass/fail/warn determination.

7.5   Environmental needs

- Basic desktop.

7.6   Special procedural requirements

The person running this test must understand the elements of an IDN table format, both as described in the reference RFCs, and in order to assess the sufficiency of a locally defined alternative and the justification for its use.

7.6.1   Requirement of LGR IDN table

An IDN table in LGR format must, besides what is stated in RFC, meet the following requirements:

- The Unicode version must be set in the "version element" in the metadata section.
- The Unicode version must be a valid Unicode version 5.2.0 or higher.
- All code points in the LGR must be included in the given Unicode version.
- The "validity-end" element (if present) in the metadata section must be in future time.

- The "validity-start" element in the metadata section must be in earlier in time than the "validity-end" element (if both are present).
- The "language" element in the metadata section must be present.
- All dispositions ("Label Generation Ruleset Dispositions") must be registered Standard Dispositions as defined in RFC 7940, section 11.3 and in the IANA registry "Label Generation Ruleset (LGR) Dispositions", https://www.iana.org/assignments/lgr-dispositions/lgr-dispositions.xhtml.

7.7   Intercase dependencies

None.

7.8   Ordered description of steps to be taken to execute the test case

1. Verify that the response to Section 2 of the IDN Self-Certification Document indicates whether the IDN table format used follows the guidelines of RFC 4290, RFC 3743 or RFC 7940.
2. If RFC 7940 IDN table is used, then all contextual rules and variant rules must be included in the "data" and "rules" elements of the LGR IDN table. Any information outside the LGR, or within the "description" element of the LGR IDN table, will be ignored if it can be included in the "data" and "rules" elements.
3. If the response to Section 2 of the IDN Self-Certification Document indicates that the IDN table is in RFC 4290 format verify that:
   a. each code point in the IDN table appears in Unicode U+nnnn notation.
   b. if the base character has any variants, the indication of its code point is followed by a VERTICAL LINE.
   c. if the base character has more than one variant, the code points for the variants are separated by a COLON.
   d. if the base character has a variant composed of a sequence of characters they are indicated with a HYPHEN MINUS between each code point.
   e. comment lines in the IDN table are preceded with a NUMBER SIGN.
4. If the response to Section 2 of the IDN Self-Certification Document indicates that the IDN table is in RFC 3743 format verify that:
   a. each code point in the IDN table appears without the Unicode U+ prefix, or if the general exception permitting the use of the Unicode "U+" prefix is invoked, that each code point in the IDN table appears in correct U+nnnn notation.
   b. if the valid code point has any variants, the columns are separated by a SEMICOLON.
   c. if there are multiple preferred or character variants, they are separated by a COMMA.
   d. if a variant is composed of a sequence of code points they are separated by a SPACE.
   e. if references are indicated, the reference number is listed in PARENTHESIS directly after the code point it and that the source is included in the list in the beginning of the IDN table.
   f. comments in the IDN table are preceded with a NUMBER SIGN.
   g. that the version number and release date are indicated.
5. If the response to Section 2 of the IDN Self-Certification Document indicates that the IDN table is in RFC 7940 format verify that:
   a. the IDN table is a well-formed XML document that conforms to the schema defined in Appendix D of RFC 7940,
   b. the IDN table meets the requirements of RFC 7940 (besides conforming to the XML schema),
   c. the IDN table meets the requirement stated in section "Requirement of LGR IDN table" (7.6.1).

Criteria for PASS:

- Section 2 of the IDN Self-Certification Document indicates the IDN table format used for the IDN table (Step 1 and 4),
- if the IDN table is in RFC 4290 format, that it conforms to those guidelines (steps under 3 above),
- if the IDN table is in RFC 3743 format, that it conforms to those guidelines (steps under 4 above),
- if the IDN table is in RFC 7940 format, that is conforms to its format and requirements (steps under 5 above).

Criteria for WARN:
- Meets the PASS criteria, and
- Some variant relation has been found to be asymmetrical or intransitive.

Criteria for FAIL:

- Section 2 of the IDN Self-Certification Document does not indicate the selected IDN table format was used,
- the conditions in Step 2-4 are not met,
- part of the information is unclear or missing.

If this test fails, none of the subsequent tests will be conducted and all will fail by default.

8. Test Case IDNvalid02: IDNA Code point validation

8.1 Test case identifier

IDNvalid02

8.2 Objective

This test verifies that the status of each tabulated code point is PROTOCOL VALID (PVALID) or CONTEXTUAL RULE REQUIRED (CONTEXTn) as defined in RFC 5892 when its algorithms are applied to an appropriate version of the Unicode Standard. The test is repeated for all IDN tables.

The IDNA Derived Property (PVALID, CONTEXTO, CONTEXTJ, DISALLOWED or UNASSIGNED) can be found for each code point on http://www.iana.org/assignments/idna-tables-6.3.0/idna-tables-6.3.0.xhtml for Unicode version 6.3. For higher version of Unicode the IDNA properties can be calculated from the Unicode database by the algorithm found in IDNA2008.

8.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
ExtendedTestTable	IDN table generated by Test Case IDNvalid02.	File
AvailableCodepointTable	A tabular listing of all PVALID and CONTEXTn code points in Unicode.	File

8.4 Outcome(s)

The outcome of the Test Case is PASS or FAIL.

8.5 Environmental needs

- Basic desktop.
- Text sorting and comparison utilities.

8.6 Special procedural requirements

None.

8.7 Intercase dependencies

IDNvalid01.

8.8 Ordered description of steps to be taken to execute the test case

For every code point in every ExtendedTestTable, extract the IDNA Derived Property value for that code point.

1. Examine ExtendedTestTable, and verify that each code point has one of the three derived IDNA property values PVALID, CONTEXTJ, or CONTEXTO.

Criteria for PASS:

- Each code point in ExtendedTestTable has one of the three derived IDNA property values PVALID, CONTEXTJ, or CONTEXTO (Step 1).

Criteria for FAIL:

- ExtendedTestTable includes one or more code points indicating IDNA property values other than PVALID, CONTEXTJ or CONTEXTO (Step 1).

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary. If this test fails, none of the subsequent tests will be conducted and all will fail by default.

9. Test Case IDNvalid03: IDNA Context rule validation

9.1 Test case identifier

IDNvalid03

9.2 Objective

This test verifies that a tabulated code point that require contextual rules can only be used according to those rules. The test is repeated for all IDN tables. This concerns:

1. Code points with the IDN property CONTEXTJ or CONTEXTO can only be used according to the contextual rule given in RFC 5892.
2. The contextual prohibitions on mixing Arabic and European digits in right-to-left labels, and on digits at the start of such labels given in RFC 5893.
3. Combining marks (non-space, spacing or enclosing marks) can never start a label (RFC 5891).
4. Required contextual rules for Modifier Letter. More on Modifier Letters below.
5. Most COMMON and INHERITED code points are usually restricted to be used with certain explicit Unicode scripts (UAX#24 and ScriptExtensions.txt of the Unicode database).
6. Code points with restricted context (e.g. must not be initial) according to Unicode.
7. All strings must be in Normalization Form C (see http://unicode.org/reports/tr15/#Norm_Forms).
8. Additional contexual rules in relevant ICANN Second-Level LGR Reference table[1], not covered above, should be included where applicable.

Each Modifier Letter (General Unicode category value Modifier_Letter, Lm) must be accompanied with a contextual rule restricting it to relevant positions or be accompanied with a statement that no such rule is needed for that Modifier Letter. If ICANN has published requirements on a certain Modifier Letter, those must be followed. Stricter rules are permitted. Each rule or non-rule should be justified with documentation showing the Modifier Letters use in the language in question (for language based IDN tables) or some language using that script (for script based IDN tables). If ICANN has published requirements or recommendation on contextual rules for a certain Modifier Letter, it is valid to refer to that document.

For many languages Second-Level LGR Reference tables are available at https://www.icann.org/resources/pages/second-level-lgr-2015-06-21-en. Besides code point repertoire and variant rules, those IDN tables contain the expected contextual rules to meet both the IDNA 2008 standards but also the security and stability requirements of IDN. Registries are welcome to use those as a reference both for LGR tables and IDN text tables.

---

[1] https://www.icann.org/resources/pages/second-level-lgr-2015-06-21-en

9.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
ExtendedTestTable	Table generated by test IDNvalid02.	File
IDNAContextualRules	The contextual rules listed in RFC 5892, Appendix A.	File, RFC 5892.
BidiDigitRules	The contextual rules given in RFC 5893, Section 2.	File, RFC 5893.
UnicodeContextRules	The contextual rules given in Unicode	Unicode (UAX#24 and ScriptExtensions.txt of the Unicode database, Normalization Form C)
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.

9.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable/warn determination.

9.5 Environmental needs

- Basic desktop.

9.6 Special procedural requirements

The person conducting this test must understand the application of the CONTEXTn rules in RFC 5892, the Bidi rule in RFC 5893, and Unicode.

9.7 Intercase dependencies

This test effectively extends into IDNvalid11.

9.8 Ordered description of steps to be taken to execute the test case

1. Examine ExtendedTestTable. If the IDN property CONTEXTO or CONTEXTJ does not appear on any row in it, the IDN table does not contain code points that can form RTL labels, and the IDN table contains no other code points requiring contextual rules, end this test as non-applicable.
2. For every code point with the IDN properties CONTEXTO or CONTEXTJ, verify that the availability is restricted as required by RFC 5892 (see Appendix A).
3. If the label is RTL as defined in RFC 5893:
   a. if it contains any code point in the range 0030..0039, ensure that no code point is in the range 0660..0669, and vice versa,
   b. if the literal component of the label consists of code points taken with the explicit script property value Arabic, ensure that no ARABIC DIGIT, EXTENDED ARABIC DIGIT, or European DIGIT is in the initial position.
4. For every code point in the General Unicode category Combining mark, verify that the availability is restricted to non-initial position.

5. For every code point in the General Unicode category Modifier Letter, verify that the availability is restricted by relevant contextual rule or accompanied with a statement that no rule is required.
6. For every code point in the General Unicode category Modifier Letter verify that supporting documentation of such use is provided; or if no contextual rule is given it is shown that unrestricted placement is permissible for that code point through supporting documentation.
7. For every code point with special property COMMON or INHERITED (except for HYPHEN-MINUS U+002D and ASCII digits 0030..0039) verify that it has the restrictions that the Unicode Standard prescribes, if any.
8. Verify that there exist contextual rules, if needed, to prevent any string to be registered unless it is in Normalization Form C.
9. Verify that applicable contextual rules found in relvevant ICANN Second-Level LGR Reference table are included in the IDN table.

For every code point besides those mentioned above, verify if the Unicode standard prescribes any restrictions, and if so, that those are followed (e.g. U+0E40).

Criteria for N/A:

- There are no code points in the IDN table with IDN properties CONTEXTO or CONTEXTJ, nor are there any other code points that require contextual rules.

Criteria for PASS:

- All code points with IDN property CONTEXTO or CONTEXTJ in the IDN tables are assigned rules that restrict them as required by RFC 5892 (step 2).
- The restrictions on digits in RTL labels required by RFC 5893 are observed (step 3).
- Other code points requiring contextual rules have appropriate rules (steps 4-6)

Criteria for WARN
- The criteria for PASS are fulfilled except that the justification for the contextual rule(s) or absence of contextual rule(s) for any Modifier Letter is missing, incomplete or unable to be validated by the testing provider. The TC will end with a warning accompanied with a comment explaining the warning. A warning is a not a failure condition.

Criteria for FAIL:

- There are code points with IDN property CONTEXTO or CONTEXTJ in one or more IDN tables that lack the rules required by RFC 5892 (step 2).
- ARABIC-INDIC digits and European digits appear together in an RTL label.
- An RTL label begins with an ARABIC-INDIC DIGIT, or an EXTENDED ARABIC-INDIC DIGIT, or a European DIGIT.
- Combining mark appears in the IDN table, but without contextual rules restricting it from initial position.
- Modifier Letter appears in the IDN table, without contextual rule restricting its use or without statement that no contextual rules are needed.
- Modifier Letter appears in the IDN table that ICANN has published with a requirement for contextual rules on that Modifier Letter and the provided contextual rule does not meet the requirement.
- Code points with property COMMON or INHERITED appear in the IDN table without contextual rules that the Unicode standard requires.

- The code point repertoire and rules make it possible to construct a string not in Normalization Form C that would be accepted.
- Applicable contextual rule from relevant ICANN Second-Level LGR Reference table is missing.
- Other code point without proper restriction.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary. If this test fails, none of the subsequent tests will be conducted and all will fail by default.

10. Test Case IDNvalid04: IDN Script validation

10.1 Test case identifier

IDNvalid04

10.2 Objective

This test verifies that the code point array in a script IDN table is restricted to a single explicit Unicode script property value as defined in the Unicode Standard Annex #24, that code points with the special script property values COMMON or INHERITED are correctly associated with the designated script, and that regardless of script property value, no code point is used in a manner alien to the designated script. The test is repeated for all IDN tables.

10.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
ExtendedTestTable	Table generated by test IDNvalid02.	File
AvailableCodepointTable	A tabular listing of all PVALID and CONTEXTn code points in Unicode, with separate columns indicating the IDN status and the Unicode script property value for the code point that keys every row. This file is provided internally.	File
ScriptIntegrityPolicies	The script integrity policies declared by the registry.	File
UAX#24	Unicode Standard Annex #24; Unicode Script Property.	File

10.4 Outcome(s)

The response to this test will be a pass/fail/warning determination.

10.5 Environmental needs

- Basic desktop.

10.6 Special procedural requirements

The person conducting this test must understand Unicode script properties designating specific scripts, as well as the values COMMON and INHERITED. These are described in UAX #24, which states that COMMON and INHERITED are assigned to code points that are used with more than one script but that this does not imply usability with all scripts. UAX #24 does not provide unequivocal guidance on how to apply such restrictions but does illustrate correct and incorrect use of those properties.

The underlying principles are to be applied in a contextually appropriate manner. For the purpose of the IDN level of the RST framework this is taken to mean that any script identifier appearing in the Unicode character name given to a COMMON or INHERITED code point must be congruent with the identifier of the IDN table being tested. For example, a Cyrillic script IDN table may not include the code point named ARABIC FATHATAN, nor would that code point be permissible in a Danish language IDN table.

Similar constraints apply to combining marks and modifier letters. Regardless of their script property values, these may not be randomly interspersed in a string. The appearance, for example, of U+0483 (COMBINING CYRILLIC TITLO) must be restricted to contexts where it is appropriate, which do not include U+047D (CYRILLIC SMALL LETTER OMEGA WITH TITLO).

The only code points with the COMMON script property that may be accepted in any IDN table are 0030..0039 DIGIT ZERO..DIGIT NINE, and U+002D HYPEN-MINUS. This is the digit and hyphen component of the basic ASCII LDH repertoire and will be referred to as "DH" in the following text. The full LDH repertoire (DH plus 0061..007A) will also be accepted if a script IDN table is primarily based on CJK Unified Ideographs or Hangul Syllables.

Any other use of COMMON or INHERITED code points in a language IDN table will require justification as being necessary to support the established orthographic practice of that language.

A script-based IDN table that indiscriminately includes all COMMON and INHERITED code points will fail.

10.7   Intercase dependencies

The outcome of this test may be contingent upon IDNvalid05. It also effectively extends into IDNvalid11.

10.8   Ordered description of steps to be taken to execute the test case

1. Examine ExtendedTestTable. If the column indicating the script property value contains the same explicit script property value for every row in the IDN table and the IDN table is labeled as supporting the designated script, end the test with pass.
2. If the special script property value COMMON appears in the IDN table and the value INHERITED does not, verify that every COMMON code point is in the DH cluster.
3. If the explicit script property value is Han, Hangul, Hiragana or Katakana, and Latin code points are included in the IDN table, verify that they are in range 0061..007A and that IDNvalid05 is pass.
4. If a code point that is not in the DH cluster has the value COMMON or if any code points have the value INHERITED, verify that the conditions discussed in UAX#24 are met.

Criteria for PASS:

- ExtendedTestTable indicates the same explicit Unicode script property value for every listed code point and the IDN table is correctly labeled as supporting that script (Step 1).
- If the special script property value COMMON appears in an IDN table and the value INHERITED does not, every code point is in the DH cluster (Step 2).
- If the explicit script property value is Han, Hangul, Hiragana or Katakana, and Latin code points in the range 0061..007A are included in the IDN table, IDNvalid05 is pass (Step 3).
- If the Unicode script property values COMMON or INHERITED appear, the conditions discussed in Section 10.6, above, are met (Step 4).

Criteria for FAIL:

- The Unicode script property values COMMON or INHERITED appear, and the conditions discussed in Section 10.6, above, are not met (Step 4).

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary.

11. Test Case IDNvalid05: IDN Script-mixing rule validation

11.1 Test case identifier

IDNvalid05

11.2 Objective

This test verifies that an IDN table including code points with more than one script property value is associated with rules that enforce the constraints on script mixing specified in the IDN Guidelines. The test is repeated for all IDN tables.

11.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
ExtendedTestTable	Table generated by test IDNvalid02.	File
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.

11.4 Outcome(s)

The response to this test will be a pass/fail/warning determination.

11.5 Environmental needs

- Basic desktop.

11.6 Special procedural requirements

None.

11.7 Intercase dependencies

This test may determine the outcome of IDNvalid04. It also effectively extends into IDNvalid11.

11.8 Ordered description of steps to be taken to execute the test case

1. Examine ExtendedTestTable. If the column indicating the Unicode script property contains only one explicit script designator and no COMMON or INHERITED code points, end this test with pass.
2. If that column contains one explicit script property value and COMMON or INHERITED code points are present, verify that they are appropriate to that script.
3. If that column contains more than one explicit script property value, verify that one of the following conditions is met:
   a. The mixing of scripts in an IDN table is restricted to LDH code points with Hangul Syllables.
   b. The mixing of scripts in an IDN table is restricted to LDH code points with Unified CJK Ideographs.
   c. The mixing of scripts in an IDN table is restricted to LDH code points with Unified CJK Ideographs intermingled with Hiragana or Katakana.

d. PolicyStatement in Section 4 of the IDN Self-Certification Document explain and justify the conditions under which the intermingling of the indicated scripts is permitted.

Criteria for PASS:

- The IDN table contains only code points with the same explicit script property value and no COMMON or INHERITED code points.
- The column indicating the Unicode script property of ExtendedTestTable contains only one explicit script property value and all listed COMMON or INHERITED code points are appropriate to that script (Step 2), or,
- The column indicating the Unicode script property of ExtendedTestTable contains more than one explicit script property value and one of the conditions in Step 3a-3d is met.

Criteria for FAIL:

- The column indicating the Unicode script property of ExtendedTestTable contains only one explicit script property value but COMMON or INHERITED code points are incongruous with the explicitly designated script (Step 2).
- The column indicating the Unicode script property of ExtendedTestTable contains more than one explicit script property value and none of the conditions in Step 3a-3d is met.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary.

12. Test Case IDNvalid06: IDN Language validation

12.1 Test case identifier

IDNvalid06

12.2 Objective

This test verifies that an IDN table associated with a language rather than a script is consistent with the script-based constraints in the preceding test cases, and that linguistic warrant is demonstrated in any policy statement permitting the intermingled use of multiple scripts in individual labels. The test is repeated for all IDN tables.

For many languages Second-Level LGR Reference tables are available at https://www.icann.org/resources/pages/second-level-lgr-2015-06-21-en. When such an IDN table is available for the languages in question it is consulted to determine if the code points in the TestTable are consistent with the language use (even if the TestTable is in text format). If the TestTable contains code points beyond those in the relevant reference IDN table, those code points will be considered before this TC is run based on the evidence of use provided.

12.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
ExtendedTestTable	Table generated by test IDNvalid02.	File
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
IDN Reference Table	IDN table found at https://www.icann.org/resources/pages/second-level-lgr-2015-06-21-en	

12.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable/warn determination.

12.5 Environmental needs

- Basic desktop.

12.6 Special procedural requirements

The person conducting this test must be familiar with basic concepts of writing systems and have access to reference material about the code point repertoires associated with the languages figuring in the PDT. Special care is needed in situations where a language uses multiple scripts but only one of them appears in a label. For example, although the Japanese writing system includes both the Latin and Katakana scripts, in a label consisting exclusively of Latin code points, U+30FC (KATAKANA-HIRAGANA PROLONGED SOUND MARK) would not be permissible.

12.7 Intercase dependencies

None.

12.8   Ordered description of steps to be taken to execute the test case

1. If TestTable is labeled as supporting a script rather than a language, end this test as non-applicable.
2. If TestTable supports a language, examine ExtendedTestTable and verify that:
   a. one explicit script property value is indicated and it is appropriate to the writing system for the designated language, and that the supported repertoire is used for that writing system. If available, an IDN Reference Table will be used to evaluate the ExtendedTestTable.
   b. more than one explicit script property value is indicated and the IDN table is declared to support a language with a writing system that uses all of those scripts, that PolicyStatement in Section 4 of the IDN Self-Certification Document provides verifiable warrant for that assertion.

Criteria for N/A:

- If TestTable is labeled as supporting a script rather than a language, end this test as non-applicable (Step 1).

Criteria for PASS:

- The code point repertoire in a language IDN table is appropriate to the writing system of the indicated language (Step 2a). Broad allowance will be made for documentable orthographic variation.
- The code point repertoire in a language IDN table is declared to support a language with a writing system that uses multiple scripts and PolicyStatement in Section 4 of the IDN Self-Certification Document provides verifiable warrant for that assertion (Step 2b).

Criteria for FAIL:

- The code point repertoire in a language IDN table is not appropriate to the writing system of the indicated language (Step 2a) or the indiscriminate inclusion of additional code points from the script(s) used for that writing system.
- The code point repertoire in a language IDN table is declared to support language with a writing system that uses multiple scripts and PolicyStatement in Section 4 of the IDN Self-Certification Document does not provide verifiable warrant for that assertion (Step 2b).

NOTE: the only writing system thus far figuring in the discussion of IDN repertoires that uses multiple scripts is Japanese, which intermingles elements of the Han, Hiragana, Katakana, and the Basic Latin scripts ("a..z"). As noted in IDNvalid04, the RST also accepts the Basic Latin repertoire together with Unified CJK Ideographs or Hangul Syllables without need for separate justification.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary.

13. Test Case IDNvalid07: IDN variant code point validation

13.1 Test case identifier

IDNvalid07

13.2 Objective

This test verifies that policies for the processing of variant relationships between listed code points are described in sufficient detail, and that all code points listed in a submitted IDN table as having variant relationships are concordant with those policies. The test is repeated for all IDN tables.

13.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
VariantAlgorithms	The IDN policies submitted for IDNvalid00 that describe the variant generation algorithms used by the registry.	File, IDN Self-Certification Document
VariantPolicies	The IDN policies submitted for IDNvalid00 that describe the variant management policies declared by the registry.	File, IDN Self-Certification Document
GRsupport	The yes/no response to the question in Section 1 of the Registry Operator's IDN Self-Certification Document regarding support for IDN at the start of General Registration.	File, IDN Self-Certification Document
EPPtags	A list of all EPP extensions needed to submit a request for the registration of an IDN label.	File

13.4 Outcome(s)

The response to this test will be a pass/fail/non-applicable/warning determination.

13.5 Environmental needs

- Basic desktop.
- EPP client.
- IPv4 or IPv6 connectivity.

13.6 Special procedural requirements

The person conducting this test must understand the concept of variant code points that ICANN applies to IDN repertoires and the associated registration policies. Further procedural constraints are discussed in Section 2.3.2 of this document. Special care needs to be taken with scripts that have single-code point and multiple-code point representations of the same character (i.e. both precomposed and combining forms, without the one canonically being replaced by the other). If both forms are included in an IDN table, variant policies must be provided to ensure that they cannot be separately delegated.

13.7 Intercase dependencies

None.

13.8 Ordered description of steps to be taken to execute the test case

1. Examine TestTable. If there is no indication of variant rules of any code point in the IDN tables, end this Test Case as non-applicable.
2. If correlations between two code points are indicated ("variant relationships"), verify that
    a. VariantPolicies in Section 4 of the IDN Self-Certification Document explains each such relationship and the constraints that attach to it, or VariantAlgorithms in Section 4 of the IDN Self-Certification Document describes the processing of each such relationship. If neither is available, end the test as failed.
    b. VariantPolicies in Section 4 of the IDN Self-Certification Document describe how the Registry Operator activates variants and that it correlates with what is stated in Exhibit A of the TLD Registry Agreement regarding the activation of variants. If there are any discrepancies, issue a warning to the Registry Operator.
3. If GRsupport is negative, the remaining steps in this test are omitted. If GRsupport is positive, proceed with the test sequence but restrict it to the IDN tables that are explicitly listed in Section 1 of the IDN Self-Certification Document and are also listed in Exhibit A of the TLD Registry Agreement.
4. If there is any uncertainty about how variant management policies are applied to an IDN table in a regard that is significant to the pass/fail determination, construct a label including a code point that is expected to be replaced by another code point upon registration and submit an EPP request for it. If the request is accepted without any indication of that transformation having been applied, end the test as failed.
5. If there is similar uncertainty about the variant management process blocking the registration of a label in one variant form if a label in another variant form has already been registered, construct a second test label in a form that should cause such blocking, and submit an EPP request for it. A further test label in a form that is not expected to be blocked may also be submitted for registration.

This EPP component of this test is only applicable to IDN tables that are listed both in Exhibit A of the Registry Agreement *and* Section 1 of their IDN Self-Certification Document.

Criteria for N/A:

- This test is not applicable to IDN tables that do not declare variant relationships between listed code points and where no such relationships are otherwise apparent (Step 1).

Criteria for PASS:

- Any IDN table that indicates variant relationships between code points must be accompanied by documentation that clearly explains how those relationships are managed in the registry (Step 2).

- If EPP tests are conducted, the registry must accept EPP requests for the registration of labels so that the behavior expected on the basis of the documentation can be verified. EPP extensions that are required in this process is included in the documentation.

- If the registry accepts and rejects test labels in accordance with the anticipated behavior, or the documentation of variant management is sufficient without EPP testing (Step 4 and 5).

Criteria for FAIL:

- The Section 4 of the IDN Self-Certification Document does not explain the variant management in sufficient detail to support the test (Step 2).
- Required EPP extensions are not included in the documentation.
- Expected EPP responses are not returned (Step 4 and 5).

14. Test Case IDNvalid09: Variant management policy

14.1 Test case identifier

IDNvalid09

14.2 Objective

This test verifies that the variant management as described in the policy is compliant with the regulation of variant management in the Registry Agreement for that TLD.

14.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
ExhibitA	Exhibit A of the Registry Agreement between ICANN and the Registry Operator for the TLD tested.	File

14.4 Outcome(s)

The response to this test will be a pass/fail/not applicable determination.

14.5 Environmental needs

- Basic desktop.

14.6 Special procedural requirements

None.

14.7 Intercase dependencies

IDNvalid07 must be completed before this test case.

14.8 Ordered description of steps to be taken to execute the test case

1. If IDNvalid07 has resulted in "not applicable" then end with N/A.
2. Examine PolicyStatement and determine what the policy for variant management is.
3. Examine ExhibitA and determine if the TLD is permitted to have any variant management.
4. Examine ExhibitA and determine if the TLD is permitted to activate any variants and what rules there are for activation.
5. Compare the results of step 3 and 4 with the result of step 2.
6. If the variant management in the PolicyStatement is compliant with the permissible variant management, end with PASS.
7. If the variant management in the PolicyStatement is NOT compliant with the permissible variant management, end with FAIL.

Criteria for N/A:

- IDNvalid07 is N/A (Step 1).

Criteria for PASS:

- PolicyStatement is compliant with ExhibitA (Step 6).

Criteria for FAIL:

- PolicyStatement is not compliant with ExhibitA (Step 7).

15. Test Case IDNvalid10: Basic IDN compliance

15.1 Test case identifier

IDNvalid10

15.2 Objective

This test verifies that the registry system is compliant with basic IDN requirements by not accepting registration of neither IDN strings in U-label format with HYPHEN in position three and four nor non-IDN ASCII strings with HYPHEN in position three and four. It also verifies that an IDN string in U-label format is rejected if it has HYPHEN in initial or final position.

15.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
GRsupport	The yes/no response to the question in Section 1 of the Registry Operator's Self-Certification Document regarding support for IDN at the start of General Registration (see also IDNvalid11).	File, IDN Self-Certification Document
EPPtags	A list of all IDN EPP extensions, such as language and script tags, needed to submit a request for the registration of an IDN label.	File

15.4 Outcome(s)

The response to this test will be a pass/fail.

15.5 Environmental needs

- Basic desktop.
- EPP client.
- IPv4 or IPv6 connectivity.

15.6 Special procedural requirements

None.

15.7 Intercase dependencies

IDNvalid11.

15.8 Ordered description of steps to be taken to execute the test case

1. Inspect PolicyStatement to determine if the TLD accepts registration of ASCII domains without any EPP/IDN extension. Unless it explicitly says that registration of ASCII domains are not supported, assume that it is.
2. The following test strings are to be registered without any EPPtag.

  a. Construct TL91 as a unique, valid ASCII label and "expect accept". TL91 is skipped if ASCII domains are not registerable.

  b. Construct TL92 as a unique, otherwise valid non-IDN ASCII label, but with HYPHEN in third and fourth positions and set "expect reject". TL92 is always included.

3. If GRsupport is negative, skip the following steps.
4. If GRsupport is positive, select one of the IDN tables that are explicitly listed in Section 1 of the IDN Self-Certification Document.
5. Determine the EPPtag used for the IDN table, if any.
6. The following test strings are to be registered as other IDN labels in IDNvalid11.

  a. Construct TL93 as a unique, otherwise valid IDN label matching all other restrictions of the selected IDN table, but with HYPHEN in third and fourth positions of the U-label.

  b. Construct TL94 as a unique, otherwise valid IDN label matching all other restrictions of the selected IDN table, but with HYPHEN in the initial position of the U-label.

  c. Construct TL95 as a unique, otherwise valid IDN label matching all other restrictions of the selected IDN table, but with HYPHEN in the final position of the U-label.

7. Submit a request to register TL91 to TL95. This is done together with the test labels from IDNvalid11, if applicable.

Criteria for PASS:

- Expected result is returned. All the following applies (only constructed labels are considered):
  a. TL91 is accepted.
  b. TL92 is rejected.
  c. TL93 is rejected.
  d. TL94 is rejected.
  e. TL95 is rejected.

Criteria for FAIL:

- Expected result is not returned. Any of the following applies (only constructed labels are considered):
  a. TL91 is rejected.
  b. TL92 is accepted.
  c. TL93 is accepted.
  d. TL94 is accepted.
  e. TL95 is accepted.

16. Test Case IDNvalid11: IDN Online registry response verification

16.1 Test case identifier

IDNvalid11

16.2 Objective

This test verifies that the online registry correctly processes test strings needed for preceding tests. The test is repeated for all IDN tables.

16.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
PolicyStatement	The IDN policies submitted for IDNvalid00.	File, IDN Self-Certification Document.
GRsupport	The yes/no response to the question in Section 1 of the Registry Operator's Self-Certification Document regarding support for IDN at the start of General Registration.	File, IDN Self-Certification Document
EPPtags	A list of all IDN EPP extensions, such as language and script tags, needed to submit a request for the registration of an IDN label.	File

16.4 Outcome(s)

The response to this test will be a pass/fail/warning determination.

16.5 Environmental needs

- Basic desktop.
- EPP client.
- IPv4 or IPv6 connectivity.

16.6 Special procedural requirements

None.

16.7 Intercase dependencies

This test is an effective extension of IDNvalid03, IDNvalid04, and IDNvalid05. It is, however, only applied to those IDN tables listed in Exhibit A of the TLD Registry Agreement that are also listed in their response to Section 1 of the corresponding IDN Self-Certification Document. If no IDN tables are listed in that section, this test is not applicable.

Test labels constructed in this test case are registered together with the test labels constructed in test case IDNvalid10.

16.8   Ordered description of steps to be taken to execute the test case

1. If GRsupport is negative, terminate this test as not applicable. If GRsupport is positive, proceed with the test sequence but restrict it to the IDN tables that are explicitly listed in Section 1 of the IDN Self-Certification Document and are also listed in Exhibit A of the TLD Registry Agreement.
2. Examine ExtendedTestTable, for every code point or group of code points that the execution of IDNvalid03 has shown to have contextual rules, construct three test labels including the code point with that property.
   a. The first test label, TL1 will place the code point in the context required by the associated rule.
   b. The second, TL2, will place the code point in a context that violates the rule.
   c. The third, TL3, will include two instances of the code point, of which one will respect the contextual rule and the other will violate it, if applicable.
3. Examine ExtendedTestTable and construct three test labels or sets of labels:
   a. The first, TL4 consists solely of listed code points.
   b. The second, TL5, includes one code point that is not listed.
   c. The third, TL6, includes at least one code point with an explicit script property value that both differs from any listed in the IDN table, and is not allowed in PolicyStatement.
4. Examine ExtendedTestTable and if it contains code points in both the range 0030..0039 and the range 0660..0669, and is an RTL label according to RFC 5893, construct a test label:
   a. that includes code points from both ranges, TL7.
5. Examine ExtendedTest, and if it includes code points from any Arabic script block construct three test labels:
   a. The first, TL8, begins with a code point in the range 0030..0039 and is followed by code points with the explicit script property value Arabic.
   b. The second, TL9, begins with a code point in the range 0660..0699 and is followed by code points with the explicit script property value Arabic.
   c. The third, TL10, begins with a code point in the range 06F0..06F9 and is followed by code points with the explicit script property value Arabic.
6. Submit a request to register (use any elements of EPPtags that may be necessary) TL1 through TL10. Include applicable test strings from IDNvalid10.

This test is only applicable to IDN tables that are listed both in Exhibit A of the TLD Registry Agreement *and* Section 1 of their IDN Self-Certification Document. Any EPP extensions required for the submission of a registration request must be included in the documentation.

Criteria for N/A:

- GRsupport is negative (Step 1).

Criteria for PASS:

- EPP extensions required for the submission of a registration request is included in the documentation.
- Expected result is returned (Step 4).
   a. TL1 is accepted.
   b. TL2 is rejected.
   c. TL3 is rejected.
   d. TL4 is accepted.
   e. TL5 is rejected.

f. TL6 is rejected.
g. TL7 is rejected.
h. TL8 is rejected.
i. TL9 is rejected.
j. TL10 is rejected.
- The expected EPP result code is returned for each test label derived from a preceding test case.

Criteria for FAIL:

- EPP extensions required for the submission of a registration request are not included in the documentation.
- Expected result is not returned (Step 4).
    a. TL1 is rejected.
    b. TL2 is accepted.
    c. TL3 is accepted.
    d. TL4 is rejected.
    e. TL5 is accepted.
    f. TL6 is accepted.
    g. TL7 is accepted.
    h. TL8 is accepted.
    i. TL9 is accepted.
    j. TL10 is accepted.
- The expected EPP result code is not returned for each test label derived from a preceding test case.

A warning will be issued if failure is not directly indicated but a qualifying remark is necessary or if the response to the request indicates that IDN registration is not yet supported in the registry.

17. Test Case IDNvalid12: Asymmetrical and intransitive variant rule verification

17.1 Test case identifier

IDNvalid12

17.2 Objective

This test verifies that any asymmetrical or intransitive variant rules do note create any security or stability issues.

Example of asymmetrical variant rules that create security and stability issues:

- Both *A* and *B* are code point that are permitted to be included in labels by an IDN table, and *A* is a variant of *B* but *B* is not a variant of *A*.

This Test Case will verify that all discovered asymmetrical or intransitive variant rules can not create security or stability issues.

17.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
VariantRules	Result from IDNvalid01, whether the IDN table contains variant rules that are asymmetrical or intransitive.	Data

17.4 Outcome(s)

The response to this test will be a pass/fail determination.

17.5 Environmental needs

- Basic desktop.

17.6 Special procedural requirements

None.

17.7 Intercase dependencies

This test is an effective extension of IDNvalid01. If that Test Case does not report of any IDN table containing any asymmetrical or intransitive variant rule then this Test Case is not applicable.

17.8 Ordered description of steps to be taken to execute the test case

1. If VariantRules is negative (no asymmetry or intransitivity), terminate this test as not applicable.
2. Examine each asymmetrical or intransitive variant rule or rule set reported by IDNvalid01.
3. If a specific variant rule or rule set can create security or stability issues, then this Test Case will conclude with FAIL.

Criteria for N/A:

- IDNvalid01 did not report any asymmetrical or intransitive variant rules.

Criteria for PASS:

- No asymmetrical or intransitive variant rule or rule set will create security or stability issues.

Criteria for FAIL:

- At least one asymmetrical or intransitive variant rule or rule set have been found to potentially create security or stability issue.

18. Test Case IDNvalid13: Pre-composed vs. decomposed character equivalence verification

18.1 Test case identifier

IDNvalid13

18.2 Objective

This Test Case verifies that the IDN table correctly handles cases where the same character can be represented either by a single code point representing a pre-composed glyph or by a sequence of code points representing a decomposed glyph. What this exactly means is illustrated in the table below.

This Test Case does not cover cases where the decomposed glyph is defined by Unicode to be in Normalization Form D, since that is already covered by Test Case IDNvalid03.

If an IDN table include all code points of the members of a pre-composed/decomposed pair, then this Test Case will verify that there are contextual rules that prevent both members to appear in the same context or that there are variant rules that handle those as "same".

Pre-composed glyph	Pre-composed code point	Decomposed glyph	Decomposed code point sequence
خٔ	0681	حٔ	062D 0654
بٔ	08A1	بٔ	0628 0654
ł	019A	ł	006C 0335

The pre-composed/decomposed pairs considered in this Test Case includes all such pairs listed in the table above, but is not limited to those.

18.3 Inputs

The following information will be needed as input for this test case:

Id	Description	Type
TestTable	The IDN table under scrutiny.	File
VariantRules	Variant rules for the IDN table	Data
ContextRules	Contextual rules for the IDN table	Data

18.4 Outcome(s)

The response to this test will be a pass/fail determination.

18.5 Environmental needs

- Basic desktop.

18.6   Special procedural requirements

None.

18.7   Intercase dependencies

None.

18.8   Ordered description of steps to be taken to execute the test case

1. If none of the IDN tables contains all code points of pre-composed/decomposed pair, then this TC will end with N/A.
2. Inspect all such pairs in the IDN table.
3. If not all such pairs are controlled by contextual rules or variant rules, as described above, then this TC ends with FAIL.
4. If all such pairs are controlled, then it ends with PASS.

Criteria for N/A:

- There are no pre-composed/decomposed pairs to verify.

Criteria for PASS:

- All pre-composed/decomposed pairs are controlled by contextual or variant rules.

Criteria for FAIL:

- At least one pre-composed/decomposed pair lacks necessary rules.

19.   General

19.1   Glossary

The glossary is available in the RST Master Test Plan.

19.2   Document change procedures

Document change procedures are documented in the RST Master Test Plan.